Power Spectral Properties of Transmission Codes(Dissertation_全文)

AUTHOR(S):

Yoshida, Susumu

CITATION:

Yoshida, Susumu. Power Spectral Properties of Transmission Codes. 京都大学, 1978, 工学博士

ISSUE DATE:

1978-05-23

URL:

https://doi.org/10.14989/doctor.r3609

POWER SPECTRAL PROPERTIES OF TRANSMISSION CODES

Susumu Yoshida

Department of Electronics
Faculty of Engineering
Kyoto University

December 1977

POWER SPECTRAL PROPERTIES OF TRANSMISSION CODES

Susumu Yoshida

Department of Electronics
Faculty of Engineering
Kyoto University

December 1977

PREFACE

This thesis deals with some spectral properties of transmission codes primarily used in baseband digital communication systems. It clarifies some relationships between the structure of an encoder and the power spectral properties of its output sequences.

Coding is one of the most important problems not only in the field of information theory but also in the field of practical communication systems. It is usually divided into two parts, i.e., a source coding and a channel coding, in order to isolate the effect of the information source in a communication system from that of the communication channel. The purpose of source coding is, roughly speaking, to represent source information by a sequence of binary digits of length as short as possible. On the contrary, channel coding is performed so that the transmitted information can be reliably reproduced at the receiving end, even in the presence of noise.

This channel coding is further divided into two parts. One is concerned with the conversion of binary sequences in a systematic way into sequences with redundancy for reliable data transmission. For instance, algebraic coding such as cyclic codes is such an example. However,it usually does not take into account how these coded sequences are modulated into electric or optical signal waveforms for transmission over actual communication media like twisted pair lines, coaxial cables or optical fiber cables.

The other is concerned with the digital data modulation which assigns one of a fixed set of waveforms to each discrete input symbol, to improve the transmission characteristics over physical communication lines. It is the study of this coding that this thesis is primarily devoted to. Most of the transmission characteristics are closely related to the power spectrum of the transmitted waveform sequence. The major goal of this coding is, thus, reshaping of power spectrum so as to match the transmission characteristics of the communication medium. We call this coding scheme and the resultant coded waveform sequence a transmission code, and various properties of transmission codes are clarified in this thesis.

Following the introductory remarks and the summary of results of this thesis in Chapter 1, Chapter 2 gives a new power spectral calculation formula of a transmission code based on a Mealy-type automaton representation of an encoder. A necessary and sufficient condition for a given transmission code to have power spectral nulls at submultiples of the pulse repetition frequency is given in Chapter 3. It is given in terms of the closed paths in the state transition diagram of an encoding automaton and seems to be the most general result with respect to spectral reshaping concerning spectral nulls. As an application of the results in Chapter 3, a catalog of encoding automata of modest size with spectral nulls at *dc* (direct current) and at half of the pulse repetition frequency is given in Chapter 4. Chapter 5 is concerned with several kinds of spectral reshapings caused by restricting the set of input sequences to an encoding automaton. As a special case, i.e., assuming that error control coded sequences are applied as inputs, a general power spectral calculation formula, applicable not only

to a unipolar code but also to a bipolar code (AMI code) is developed in Chapter 6. Chapter 7, slightly different from the preceding chapters, gives the general description of an optically linked laboratory computer network LABOLIN developed in Professor Yajima's research laboratory at Kyoto University, followed by the description of specific network components and a simple encoding scheme of a Fibonacci code, developed by the author. Concluding remarks are given in Chapter 8.

CONTENTS

LIST OF SYMBOLS

$A=(a_{ij})$	state transition matrix (adjacency matrix)
$\{a_n\}$	digital message sequence
$\mathbf{C}$ ($\mathbf{C}^{(r)}$)	codeword vector (r-th codeword vector)
$c_i^{(r)}$	i-th digit of r-th codeword
D	finite set of consecutive integers
f_r	pulse repetition frequency
$f(n)$	$\{1-(1-2p)^n\}/2$
$g(t)$	standard unit pulse
$G(f)$	Fourier transform of $g(t)$
$\mathbf{H}$	parity check matrix
$\mathbf{H}'$	(modified) parity check matrix
$\mathbf{h}_i$	i-th row vector of $\mathbf{H}'$
h_i	Hamming weight of $\mathbf{h}_i$
$\mathbf{I}$	column vector consisting of information digits
m	mean of output symbols
m_I	local average (in Q_I)
M	finite automaton
M_c	composite finite automaton
N	period of state transition diagram ; block length
$N_R(n)$	number of allowed input sequences of length n satisfying the given restriction
P	state transition probability matrix
p	occurrence probability of input symbol 1
Q_I	subset of S
$R(n)$	autocovariance of a sequence $\{a_k\}$
$\overline{R}(n)$	autocovariance of a sequence $\{a_k-m\}$
S	finite set of states
$S(f)$	normalized power spectrum

$S_c(f)$	continuous component of $S(f)$
$s.t.$	such that
T	pulse interval
t	transposition of matrix ; time
u_n	Fibonacci series
$w(f)$	power spectrum
$w_c(f)$	continuous component of $w(f)$
$w_d(f)$	discrete component of $w(f)$
$x(t)$	digital pulse sequence
X	finite set of input symbols
Y	finite set of output symbols
Z	output matrix
δ	next state function
$\delta_d(f)$	Dirac's delta function
η	limiting transmission efficiency
$\eta_B(n)$	η in the case of block coding of length n
$\mathfrak{1}$	$(1\ 1\ 1\ \ldots\ 1)^t$
λ	output function
λ_{max}	largest positive eigenvalue of matrix A
ω	$2\pi f$
$\oplus$	mod 2 addition
$\wedge$	logical AND
*	elementwise multiplication of matrices

CHAPTER 1 INTRODUCTION

After presenting introductory remarks on power spectral properties of transmission codes together with historical review of research efforts concerning transmission codes, main results obtained in this thesis are summarized in this chapter.

1.1 Introductory Remarks on Transmission Code and Its Power Spectrum

With the growing needs for reliable and large capacity digital data transmission systems, much attention has been paid toward the coding of digital data. Of various coding processes, transmission codes have considerable significance in transmitting digital data over practical communication channels. In fact, there arise many problems of engineering interest when transmitting digital data, such as reduction of *dc* (direct current) wandering, suppression of intersymbol interference and inclusion of self-clocking capability. A transmission code is an encoding scheme of digital data for transmission whose main purpose is to overcome such problems. As a magnetic recording system can be also regarded as a kind of communication system, where storage medium corresponds to communication channel, the same discussion applies. With the recent advent of various superior communication media like large

capacity coaxial cables and optical fiber cables, the development of transmission codes for efficient and reliable data transmission is desired more than ever before.

The simplest example of a transmission code is a unipolar code, in which input symbol 1 corresponds to the presence of a pulse, whereas 0 corresponds to the absence of a pulse. One of the most widely used codes is a bipolar code (also referred as AMI - alternate mark inversion - code), which was introduced for the multiplex transmission of telephone signals by PCM (pulse code modulation) on cables. This code is now an almost universal standard and is characterized by the rule that input symbol 0 corresponds to the absence of a pulse, whereas input symbol 1 corresponds to the presence of a pulse whose polarity changes alternately. Many other transmission codes have thus far been developed, each of which has respective merits and demerits. Possibly, many other transmission codes could be developed, which exhibit better characteristics for some transmission lines. Consequently, it is important to use a transmission code that matches the specific requirements of a given communication line.

However, transmission codes have been, thus far, chosen in a rather heuristic manner based on experience. So, it has been strongly desirable to develop a systematic method for constructing a transmission code with specific properties.

The most important aspect characterizing a transmission code is the power spectral density, more commonly called the power spectrum. It generally represents the power distribution of the ensemble of transmitted signals corresponding to the digital coded sequences. That is,

the power spectrum $w(f)$ means that approximately the total power of $w(f)\Delta f$ exists between frequencies f and $f+\Delta f$. Thus the power spectrum is very useful when analyzing the transmission characteristics of a digital data transmission system in the frequency domain. How to calculate the power spectrum of a given transmission code will be stated in Chapter 2.

Aiming at systematic construction of transmission codes with specific properties, we are interested in clarifying the relationship between the spectral properties and the encoding scheme, as well as in a calculation method of the power spectrum.

In this thesis, the relation between the digital coded sequence with power spectral nulls at $f=(h/k)f_r$ (h: nonnegative integer, k: positive integer, f_r: pulse repetition frequency) and the structure of the encoder, regarded as a finite automaton, is studied. A necessary and sufficient condition for the encoding automaton to have such a property will be given in terms of the closed paths of its state transition diagram.

Other spectral properties of transmission codes which are worthy of consideration are also treated in this thesis. That is, when we simply say power spectrum of transmission code, we usually mean the power spectrum of digital coded signal generated by the corresponding encoding scheme assuming independent random input sequences. Sometimes, however, input sequences for an encoder are not necessarily independent, but rather correlated by some constraint imposed on them. This leads to the reshaping of the power spectrum of a transmission code. Thus, how the power spectrum of a transmission code is reshaped when the set of possible input sequences is restricted to its proper subset is discussed.

Moreover, as a typical example with correlated inputs, we will consider the power spectral calculation method for error detecting and correcting codes. Recently, in many situations where reliable communication is desired, error detecting and correcting codes are being put into wide use. The power spectrum of the output digital signal for independent input sequences is generally different from that for error detecting or correcting coded sequences . Therefore, it is very important in designing data transmission systems to know how the power spectrum alters when such an encoding is performed.

In summary, this thesis is devoted primarily to the study of spectral properties of transmission codes. In most of the chapters, the study is carried out by regarding the transmission code as the output digital sequence of a finite automaton. In this case, research on the spectral properties of transmission codes is reduced to the investigation of the structure of encoding automaton, which is more tractable due to the well developed finite automata theory and Markov chain theory.

The main results obtained in this thesis will be summarized in Section 1.3.

1.2 Background and Historical Review

Many research efforts have been directed toward the development of transmission codes with desirable properties for a given transmission line. A brief historical review about transmission codes is given in this section [1-2].

As described in the preceding section, the simplest example of a transmission code is the unipolar code. One of the most widely used code is the bipolar code [3], which is now an almost universal standard. Many useful transmission codes can be derived from it by making slight modifications. For instance, B6ZS (bipolar-with-six-zero-substitution) code, HDB (high density bipolar) code, CHDB (compatible high density bipolar) code are derived by replacing a long consecutive sequence of zeros by a special sequence which is decodable without ambiguity [1]. While, a count-n bipolar code [3] is obtained by adopting the rule such that n successive ones are grouped to have the same polarity while the polarity changes alternately between successive groups. A hold-n bipolar code [4] is obtained by changing the polarity of input symbol one only when input symbol zero occurs or input symbol one continues more than n time slots. Both count-1 and hold-1 codes reduce to the usual bipolar code.

Another class to which many transmission codes belong is alphabetic codes. By the designation "alphabetic codes" we mean those codes whose digits are grouped in "characters" of certain number of consecutive digits, encoding being done one character at a time. The most publicized of these codes is the Paired Selected Ternary (PST) code [5], in which a group of two bits is represented

by a group of two ternary digits. The purpose of this type of encoding is to simultaneously eliminate *dc* and the ternary character 00. There are two possible ternary characters encoding the binary combinations 01 and 10, and *dc* is balanced by alternating the two kinds of optional characters.

Partial response code is another interesting code that is used in many recent data sets. It is generated by binary signal elements which interfere with each other. Due to this controlled amount of intersymbol interference, it is possible to shape the system spectrum, for instance to place nulls in the frequency response. Lender [6-7] first introduced duobinary partial response signaling in a data transmission system. Kretzmer [8] listed several important partial response signalings and compared their transmission characteristics. The input and output relation of a partial response channel can be expressed by a state transition diagram or a finite automaton. From this standpoint, a more systematic treatment of a communication channel with memory has been made and the theory of partial response signaling has been extended by several authors: Kanaya [9], Miyakawa and Harashima [10], and Kambayashi and Yajima [11]. These kinds of codes tend to have spectral nulls at submultiples of the pulse repetition frequency if the quantity of intersymbol interference is properly chosen. Consequently, they constitute a subclass of transmission codes discussed in this thesis, if we regard the input and output relation of the total communication system including channels with memory as the encoding scheme. Some other more recent discussions about partial response signaling are summarized in [12].

On the other hand, little attention seems to have been given to the systematic treatment of a transmission code itself before Kanaya's work [13]. He suggested that a transmission code can be treated systematically by regarding it as a deterministic Moore-type finite automaton or as a sequence transducer with random input. He also demonstrated the possibility that some statistical properties might be derived from the state transition diagram of a finite automaton.

Following him, Yasuda and Inose [14-16] successfully applied the automata theoretic approach to the calculation of power spectra of transmission codes and obtained a simple and elegant formula. Compared with previous calculation methods such as Bennett's, their formula is excellent in that it is not only more systematic but also given in a closed form, not in an infinite series where a truncation error is inevitable.

They also proved the interesting result that the *dc* (direct current) spectral component of a transmission code disappears if and only if it is generated by a loop-sum-zero state transition diagram, i.e., a state transition diagram in which for any closed path the algebraic sum of the outputs along it is always equal to zero [17-18]. This result is a very useful clue for constructing a balanced (without *dc*) transmission code systematically. However, nothing seems to have been known about the more general condition that the power spectrum disappears at the frequency $f=(h/k)f_p$, except for a special case of an alphabetic code [19].

G.L.Cariolaro and G.P.Tronca [20] have recently developed a power spectral calculation method applicable to a block code based on an automata theoretic approach. However, it is effectively applicable only when the num-

ber of states of the encoding automaton is relatively few. Thus, it is impractical to apply it to the calculation of power spectra of error control codes, where the number of states of the encoding automaton grows rapidly as the block length increases.

More recently, however, Gilchrist and Thomas [21] showed a power spectral calculation method applicable to error control codes based on Bennett's formula. It is the first approach to the power spectral calculation of error control codes. However, as their formula only covers the calculation of simple PAM (pulse amplitude modulation) coded error control code, where the amplitude is directly proportional to the input symbol , it is impossible to calculate, for instance, the power spectrum of bipolar coded error control codes. Thus, a more general formula applicable to a wider range of transmission codes is desirable.

1.3 Summary of Results

In this thesis, extending the research results mentioned in Section 1.2 and, at the same time, pursuing further theoretical development, we have studied power spectral properties of transmission codes, aiming at generalization of the relationships between the structure of encoding automaton and its power spectral properties. Main results obtained in this thesis are summarized in the following.

(1) An efficient power spectral calculation procedure is developed by regarding an encoder as a Mealy-type finite automaton. It is a rather compact algorithm than that already developed based on a Moore-type finite automaton and is described in Chapter 2.

(2) A necessary and sufficient condition for a transmission code to have a power spectral null at frequency $f=(h/k)f_r$ is obtained in terms of the structure of the encoding automaton, where h and k are nonnegative and positive integers, respectively, and f_r is the pulse repetition frequency. This general result for the problem of power spectral nulls is sure to become a useful guideline toward a systematic construction of encoders with specified properties. It is discussed in detail in Chapter 3.

(3) Based on the results mentioned above (2), a catalog of pseudoternary transmission codes with a spectral null at $f=0$ or $(1/2)f_r$ is obtained in the form of a Mealy-type state transition diagram with up to four states under the assumption of specific input/output symbols assignment. This catalog seems to have significant practical importance in that it exhaustively lists all possible transmission codes satisfying the specified requirements, although it is only the first step toward a systematic construction of transmission codes. Details of the catalog are discussed in Chapter 4.

(4) We have shown that the power spectra of various transmission codes can be reshaped by only imposing a certain restriction on the input sequences. Particularly, generation of power spectral nulls or discrete line spectral components is shown to be possible by slightly changing the statistics of input sequences. Both the calculation formula for limiting transmission efficiency

and the encoding rule attaining the limit with a certain input restriction are also developed. Details are discussed in Chapter 5.

(5) A power spectrum calculation procedure for error control codes such as a multiple error correcting cyclic code is developed. Since an error control code generally requires too many states to represent it in a finite automaton, it is not suitable in this case to apply the result (1) directly. So, a new systematic procedure primarily based on row operations of the parity check matrix corresponding to an error control code is developed. It is actually applied to evaluate the power spectrum of a bipolar coded error control code, which has never been calculated before, as well as that of unipolar coded one. Chapter 5 treats it in detail.

(6) An optically linked laboratory computer network LABOLINK, which has a variety of features, has been developed at Professor Yajima's research laboratory. In LABOLINK, state-of-the-art technologies such as fiber-optic data transmission have been introduced to show their effectiveness. The author has taken part in the project and developed a kind of digital multiplexer called a "single-line multi-connector" and an optical fiber data transmission system as well as some other parts of LABOLINK. A single optical fiber transmission line, if coupled with a single-line multi-connector, can be utilized for multi-purpose simultaneous data transmission. Part of LABOLINK and a simple encoding scheme of Fibonacci code utilized in the single-line multi-connector developed by the author, together with the overall outline of LABOLINK, are presented in Chapter 6.

CHAPTER 2 CALCULATION FORMULA OF POWER SPECTRA OF TRANSMISSION CODES

In this chapter, power spectral calculation formula of a transmission code is described. After presenting a formula derived by W.R.Bennett [22] which is given in an infinite series, a closed form formula based on an encoding automaton will be stated. Although H.Yasuda and H. Inose [14,16] developed a formula directly applicable to a Moore-type automaton, we will show here one directly applicable to a Mealy-type automaton.

2.1 Introduction to Power Spectral Calculation Formula

In this thesis, we shall treat a transmission code whose output sequence is expressible by

$$x(t) = \sum_{n=-\infty}^{\infty} a_n g(t-nT) \quad , \tag{2.1}$$

where the sequence a_n, n being integer and going from $-\infty$ to ∞, represents the message values and $g(t)$ and T denote a fixed standard unit pulse and pulse interval, respectively. In this section, we shall describe how to calculate the power spectrum of pulse sequence of expression (2.1). Since $g(t)$ is fixed, the message sequence $\{a_n\}$ determines (2.1) completely, where $\{a_n\}$ represents an

infinite message sequence ..., a_{-n},..., a_{-1}, a_0, a_1,..., a_n,.... Thus, statistics of message sequence $\{a_n\}$ will determine power spectrum of pulse sequence $x(t)$. However, message sequence $\{a_n\}$ is merely a typical sample of all possible message sequences. Thus, when we calculate power spectrum of transmission code, we must take the ensemble of possible pulse sequences into account.

It is assumed that the message ensemble is ergodic and stationary. Mean m is defined by the ensemble average:

$$m = \mathrm{av}(a_n) \quad . \tag{2.2}$$

Autocovariance $R(n)$ is defined by the ensemble average:

$$R(n) = \mathrm{av}(a_k a_{k+n}) \quad . \tag{2.3}$$

Then, the statistics of digital message ensemble will be related to those of the pulse sequence ensemble.

Generally, the power spectrum of the coded sequence (2.1) is divided into discrete line spectrum $w_d(f)$ and continuous spectrum $w_c(f)$. The former is considered first. That is, we evaluate the ensemble average of $\{x(t)\}$ by holding t fixed and averaging over all members.

$$\begin{aligned} \mathrm{av}\{x(t)\} &= \sum_{n=-\infty}^{\infty} \mathrm{av}\{a_n\} g(t-nT) \\ &= m \sum_{n=-\infty}^{\infty} g(t-nT) \quad . \end{aligned} \tag{2.4}$$

Since $\mathrm{av}\{x(t+T)\} = \mathrm{av}\{x(t)\}$, Fourier series expansion is possible. If we let $G(f)$ be the Fourier transform of $g(t)$, then we obtain

$$\mathrm{av}\{x(t)\} = m f_r \sum_{n=-\infty}^{\infty} G(n f_r) \exp(2n\pi j f_r t) \quad . \tag{2.5}$$

where $f_r = 1/T$ is the pulse repetition frequency. Thus, discrete line spectrum $w_d(f)$ of power spectrum is calculated as

$$w_d(f) = m^2 f_r^2 |G(f)|^2 \sum_{n=-\infty}^{\infty} \delta_d(f-nf_r) \quad , \tag{2.6}$$

where $\delta_d(f)$ is Dirac's delta function (conventional $\delta(f)$ is not used here because δ will be used as a next state function of a finite automaton later).

Next, the remainder part of pulse ensemble is considered, i.e.,

$$\{y(t)\} = \{x(t)\} - \mathrm{av}\{x(t)\} = \sum_{n=-\infty}^{\infty} (a_n - m) g(t-nT) \quad , \tag{2.7}$$

Then $\mathrm{av}\{y(t)\} = 0$.

Bennett derived a convenient formula for continuous power spectrum as follows [22].

First , consider the ensemble $y_N(t)$ which includes only the pulses from $n=-N$ to N:

$$y_N(t) = \sum_{n=-N}^{N} (a_n - m) g(t-nT) \quad , \tag{2.8}$$

then the Fourier transform $Y_N(f)$ of $y_N(t)$ is given by

$$Y_N(f) = \sum_{n=-N}^{N} (a_n - m) G(f) \exp(-2n\pi jfT) \quad . \tag{2.9}$$

Then, by using Parseval's theorem and averaging the ensemble $y_N^2(t)$ over the interval $-NT$ to NT, which is written as $\mathrm{av}_N\, y_N^2(t)$, he derived the following relations,

$$\begin{aligned} \mathrm{av}_\infty y_\infty^2(t) &\triangleq \lim_{N\to\infty} \mathrm{av}_N y_N^2(t) \\ &= \lim_{N\to\infty} \int_{-\infty}^{\infty} \frac{\mathrm{av}\,|Y_N(f)|^2}{(2N+1)T} df = \int_{-\infty}^{\infty} w_c(f)\, df \end{aligned}$$

$$\text{where} \quad w_c(f) \triangleq \lim_{N\to\infty} \frac{\mathrm{av}\,|Y_N(f)|^2}{(2N+1)T} \quad . \tag{2.10}$$

The $w_c(f)$ thus defined satisfies the requirement for a

spectral density. Substituting Eq.(2.9) into Eq.(2.10) and performing necessary manipulations, we can obtain the following well-known calculation formula for continuous power spectrum, sometimes called Bennett's formula.

$$w_c(f)=f_r|G(f)|^2\{R(0)-m^2+2\sum_{k=1}^{\infty}(R(k)-m^2)\cos 2\pi kfT\}. \quad (2.11)$$

The total power spectrum $w(f)$ is then defined by

$$w(f) = w_c(f) + w_d(f) . \quad (2.12)$$

If we change the averaging interval of the ensemble $y_N^2(t)$ in the above, then another equivalent representation of $w_c(f)$ is obtained:

$$w_c(f) = \lim_{\substack{N_1\to-\infty \\ N_2\to\infty}} \frac{|G(f)|^2}{(N_2-N_1)T} \mathrm{av}\left|\sum_{n=N_1}^{N_2-1}(a_n-m)\exp(-2n\pi jfT)\right|^2. \quad (2.13)$$

This will be used in the subsequent chapter.

However, this is not true when treating a more general case , sometimes described as "periodically stationary", where the ensemble statistics varies periodically with time. The name "cyclostationary" is also used for this type of process [22]. The error detecting or correcting block codes, for instance, are regarded as typical examples of cyclostationary sequences with a period equal to the block length. In this case, we cannot apply the Bennett's formula directly. A slight modification will be needed to derive a similar formula for cyclostationary process. Moreover, a little preparation is necessary beforehand in order to describe the formula concisely and accurately. That is, in case of cyclostationary process, it can be completely characterized by an encoding automaton (details are discussed in the next chapter), whose state transition diagram is periodic with period greater than unity. If we let the

period be N, then the whole set of states S of the encoding automaton can be divided into N mutually disjoint subsets Q_I (I=0, 1, 2,..., N-1) such that $S = Q_0 \cup Q_1 \cup \cdots \cup Q_{N-1}$, and $Q_I \cap Q_J = \phi$ ($I \neq J$). In addition, any state in Q_I will surely make a state transition to some state in $Q_{I+1(mod N)}$ for any I, which is illustrated in Fig.2.1. Then, the author introduced a very useful notion of 'local average' of an encoding automaton. It is defined as the ensemble average of the output values not over the whole set of states but over a single subset of states Q_I for some I, and we say that local average in Q_I is m_I if $m_I = \underset{Q_I}{\mathrm{av}} \{a_i\}$.

If we restate the calculation formula described in [16] using local averages, the following formula is obtained. That is,

$$w_d(f) = f_r^2 |G(f)|^2 \sum_{k=-\infty}^{\infty} \delta_d(f - \frac{k}{N} f_r) \left| \frac{1}{N} \sum_{i=0}^{N-1} m_i e^{-j2\pi k \frac{i}{N}} \right|^2, \quad (2.14)$$

$$w_c(f) = f_r |G(f)|^2 \{\overline{R}(0) + 2 \sum_{k=1}^{\infty} \overline{R}(k) \cos 2\pi k f T\}, \quad (2.15)$$

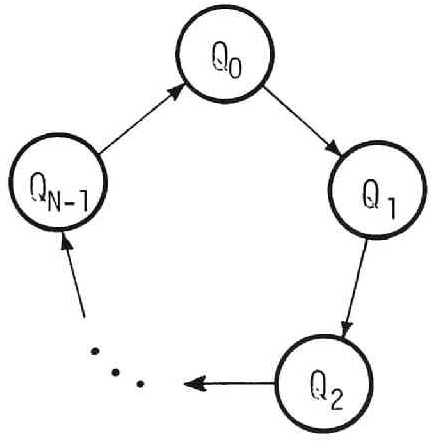


Fig.2.1 Partition of state space of a state transition diagram with period N.

where $\bar{R}(k) = \mathrm{av}\{(a_i - m_i)(a_{i+k} - m_{i+k(mod N)})\}$

$$= R(k) - (1/N)\sum_{i=0}^{N-1} m_i m_{i+k} \quad . \qquad (2.16)$$

Thus, if we replace $R(k)-m^2$ by $\bar{R}(k)$ for $k = 0,1,2,\ldots$, then Bennett's formula can be again applied to evaluate continuous power spectrum of cyclostationary process. On the other hand, discrete line spectrum is represented concisely by using local averages. For example, if we assume the period be four and the corresponding local averages be m_0, m_1, m_2 and m_3, then we obtain

$$w_d(nf_r) = \frac{1}{16} f_r^2 |G(nf_r)|^2 (m_0+m_1+m_2+m_3)^2,$$

$$w_d((n+\tfrac{1}{2})f_r) = \frac{1}{16} f_r^2 |G((n+\tfrac{1}{2})f_r)|^2 (m_0-m_1+m_2-m_3)^2,$$

$$w_d((n\pm\tfrac{1}{4})f_r) = \frac{1}{16} f_r^2 |G((n\pm\tfrac{1}{4})f_r)|^2 \{(m_0-m_2)^2+(m_1-m_3)^2\},$$

where delta function is not explicitly shown.

2.2 Calculation of Power Spectrum Based on Mealy-type Encoding Automaton

A power spectral calculation method directly applicable to a Mealy-type encoding automaton is presented in this section. Though an efficient power spectrum calculation method based on a Moore-type encoding automaton has been already developed by H.Yasuda and H.Inose [14,16], it is not straightforward to apply it to a

Mealy-type automaton. The method described here, however, has the feature that it is directly applicable to a Mealy-type encoding automaton , a more compact representation than a Moore-type one. Therefore it requires much less storage area and computation time for matrix operations needed almost exclusively in the calculation process. It was developed by trying to remove the redundancy inherently existing in the representations of most of matrices in their method. However, the derivation process of our closed form formula is almost the same as that of their method. The calculation formula is given for the general case where an encoding automaton is strongly connected and the corresponding state transition diagram has a period N.

A Mealy-type encoding automaton is represented as a quintuple $M = (S,X,Y,\delta,\lambda)$, where $S=\{s_1,s_2,\ldots,s_n\}$ is a finite set of states, $X=\{0,1\}$ is a set of binary input symbols, Y is a set of output symbols, δ is a next state function $\delta:S\times X\rightarrow S$, and λ is an output function $\lambda:S\times X\rightarrow Y$. If a state transition diagram of M has a period N, then its state space S can be divided into mutually disjoint N subsets as illustrated in Fig.2.1. In the special case of $N=1$, reduces the state transition diagram to the aperiodic case producing ergodic pulse sequences.

Now several matrices and vectors needed in the following discussion are defined. By rearranging the order of the states properly, the state transition probability matrix $P=(p_{ij})$ and output matrix $Z=(z_{ij})$ are represented in the following form:

$$P=\begin{array}{c} \\ Q_0 \\ Q_1 \\ \vdots \\ \\ Q_{N-1} \end{array}\begin{bmatrix} \mathbf{0} & P_{12} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & P_{23} & & \vdots \\ \vdots & & & \ddots & \vdots \\ \mathbf{0} & & & & P_{N-1,N} \\ P_{N1} & \mathbf{0} & \cdots & & \mathbf{0} \end{bmatrix}, \quad Z=\begin{bmatrix} \mathbf{0} & Z_{12} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Z_{23} & & \vdots \\ \vdots & & & \ddots & \vdots \\ \mathbf{0} & & & & Z_{N-1,N} \\ Z_{N1} & \mathbf{0} & & & \mathbf{0} \end{bmatrix}$$

where p_{ij} and z_{ij} are defined as follows:

$$p_{ij}=\begin{cases}\Pr(x) & \text{if } \delta(s_i,x)=s_j\\ 0 & \text{otherwise}\end{cases},\quad z_{ij}=\begin{cases}\lambda(s_i,x) & \text{if } \delta(s_i,x)=s_j\\ 0 & \text{otherwise,}\end{cases}$$

and $\Pr(x)$ is occurrence probability of an input symbol x. If it happens that two different input symbols cause the same state transitions, then a slight modification is needed, which will be referred later.

The matrix P^N defines the state transition probabilities observed at every N time units, namely from Q_I to itself for any I. If we let $P_N^\infty \triangleq \lim_{k\to\infty}(P^N)^k$, then each square submatrix P_{II}^∞ of P_N^∞ shown below consists of the same row vectors $\mathbb{P}_I$ representing stationary state occupancy probabilities in Q_I. Furthermore, we let the

$$P^N=\begin{bmatrix}P_{11} & \mathbf{0} & & \mathbf{0}\\ \mathbf{0} & P_{22} & & \\ & & \ddots & \\ & & & \mathbf{0}\\ \mathbf{0} & & & P_{NN}\end{bmatrix},\quad P_N^\infty=\begin{bmatrix}P_{11}^\infty & \mathbf{0} & & \mathbf{0}\\ \mathbf{0} & P_{22}^\infty & & \\ & & \ddots & \\ & & & \mathbf{0}\\ \mathbf{0} & & & P_{NN}^\infty\end{bmatrix},\quad P_{II}^\infty=\begin{bmatrix}\mathbb{P}_I\\ \mathbb{P}_I\\ \vdots\\ \mathbb{P}_I\end{bmatrix}$$

row vector in P_N^∞ corresponding to the states in Q_I be $\mathbb{P}_{SVI}$ and let $\mathbb{P}_{SV} = (\sum_{I=1}^{N}\mathbb{P}_{SVI})/N$, then $\mathbb{P}_{SV}$ is the stationary state occupancy probability vector over the whole set of states S. An elementwise multiplication of matrices are defined and denoted as $Q=P*Z$ where $q_{ij}=p_{ij}z_{ij}$. In the special case where two input symbols cause the same state transitions, q_{ij} must be replaced by $\sum p_{ij}z_{ij}$ where summation is taken over all the input symbols associated with the state transition from state i to j.

[Lemma 2.1] Define autocovariances $R(k)$ and $\overline{R}(k)$ of the output sequences $\{a_i\}$ and $\{a_i\text{-av}(a_i)\}$, respectively, as

follows:

$$R(k)=\mathrm{av}\{a_i a_{i+k}\}$$

$$\overline{R}(k)=\mathrm{av}\{(a_i-m_i)(a_{i+k}-m_{i+k})\},$$

then, $R(k)=\boldsymbol{\pi}_{sv} Q P^{k-1} Q \boldsymbol{\eta}$ (for $k\geq 1$)

$$\overline{R}(k)=\boldsymbol{\pi}_{sv} Q (P^{k-1}-P_N^{\infty} P^{k-1}) Q \boldsymbol{\eta} \quad (\text{for } k\geq 1)$$

$$\overline{R}(0)=\boldsymbol{\pi}_{sv}(Q*Z)\boldsymbol{\eta} - (1/N)\sum_{i=0}^{N-1} m_i^2 ,$$

where $Q=P*Z$, $\boldsymbol{\eta}=(1,1,\ldots,1)^t$ (t: transposition), $m_i=\mathrm{av}(a_i)$: ensemble average of a_i for fixed i and $m_i=m_j$ ($i=j \bmod N$).

(Proof) Compared with Moore-type automaton, it is each state transition or directed edge of a state transition diagram that an output symbol is associated with in the case of Mealy-type diagram. On the other hand, $R(k)$ is defined by the ensemble average of the product of the two output symbols separated by k time units. Referring to Fig.2.2, the probability that z_{ij} is the first output in

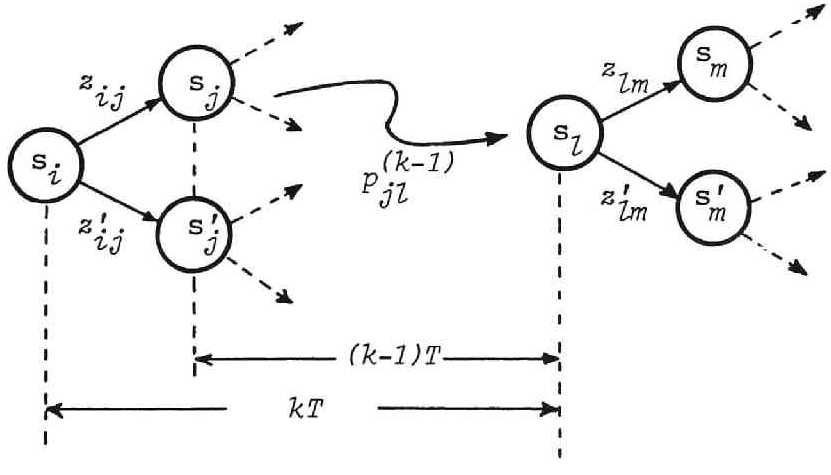


Fig.2.2 Illustration of state transitions of a Mealy-type automaton separated by k time units.

s_i followed by z_{lm} in s_l after $(k-1)$ time units is $p_i(p_{ij}z_{ij})p_{jl}^{(k-1)}(p_{lm}z_{lm})$, where p_i is the entry of $\mathbb{P}_{sv}$ corresponding to s_i and $p_{jl}^{(k-1)}$ is the (j,l) entry of P^{k-1}. Taking the summation with respect to all the states, we obtain

$$R(k)=\sum_{ijlm} p_i(p_{ij}z_{ij})p_{jl}^{(k-1)}(p_{lm}z_{lm})=\mathbb{P}_{sv}QP^{k-1}Q\,\eta \quad (k\geq 1).$$

Moreover, due to the fact that $\mathbb{P}_I P_{I,I+1}=\mathbb{P}_{I+1}$, we can get the following equation,

$$\begin{aligned}\underset{i}{\mathrm{av}}\{m_i m_{i+k}\}&=(1/N)\sum_{ijlm}(Np_i)(p_{ij}z_{ij})(Np_l)(p_{lm}z_{lm})\\&=\sum p_i(p_{ij}z_{ij})(Np_l)(p_{lm}z_{lm})=\mathbb{P}_{sv}QP_N^{\infty}P^{k-1}Q\,\eta \quad (k\geq 1).\end{aligned}$$

Accordingly, the next equation can be derived,

$$\bar{R}(k)=\mathbb{P}_{sv}Q(P^{k-1}-P_N^{\infty}P^{k-1})Q\,\eta \qquad (k\geq 1).$$

In the special case where k equals to zero, we obtain

$$\begin{aligned}\bar{R}(0)&=\mathrm{av}\{(a_i-m_i)^2\}=\mathrm{av}\ a_i^2-\mathrm{av}\ m_i^2\\&=\sum_{i,j} p_i p_{ij} z_{ij}^2-(1/N)\sum_{i=0}^{N-1} m_i^2\\&=\mathbb{P}_{sv}(Q*Z)\eta-(1/N)\sum_{i=0}^{N-1} m_i^2 .\end{aligned}$$

[Proposition 2.1] The continuous power spectrum of a transmission code with period N is calculated by the following equation:

$$w_c(f)=f_r|G(f)|^2 S(f),$$

$$S(f)=\bar{R}(0)+2\,\mathbb{P}_{sv}QV^{-1}\sum_{i=1}^{N}P^{i-1}\{(I-P_N^{\infty})\cos i\omega T-R\cos(N-i)\omega T\}Q\,\eta$$

where $S(f)$: normalized power spectrum, $\bar{R}(0)=\mathbb{P}_{sv}(Q*Z)\eta-(1/N)\sum_{i=0}^{N-1} m_i^2=\mathbb{P}_{sv}(Q*Z-QP_N^{\infty}P^{N-1}Q)\eta$, $V=I-2R\cos N\omega T+R^2$,

$R=P^{N}-P_{N}^{\infty}$ and I: an identity matrix.

(Proof) Since the transformation process, which leads to the closed form formula from the Bennett's infinite series, is almost the same as that stated in [16] in detail, we will omit it here.

CHAPTER 3 TRANSMISSION CODES WITH POWER SPECTRAL NULLS AT SPECIFIC FREQUENCIES

This chapter presents the result of an investigation on the relationship between a transmission code with power spectral nulls at several specific frequencies and its encoding automaton. This general result is sure to become a useful guideline toward a systematic construction of encoders with power spectral nulls.

3.1 INTRODUCTION

As is previously discussed, a transmission code has been hitherto selected in a rather heuristic manner, based upon experience. Recently, however, it was pointed out that an encoder for a transmission code is well characterized by a finite automaton model [13]. From this stand point, H.Yasuda and H.Inose have shown that any code generated by a loop-sum-zero transition diagram, defined in [17], has no *dc* component [17] and proven, in their later paper [18], that the converse is also true. Nothing has been published on the conditions for the more general case, where power spectrum disappears at frequency $f=(h/k)f_r$, except for a special case of an alphabetic code [19], where h and k are nonnegative and positive integers, respectively.

This chapter presents a necessary and sufficient

condition for a given code to have a power spectral null at frequency $f=(h/k)f_r$. This general result will be stated in terms of closed paths in the state transition diagram of an encoder. These codes are important because they are inherently modulating, i.e., their signal energy is concentrated into a predetermined frequency range. They can, therefore, match specific requirements in the design of transmission systems. Furthermore, a code with spectral null at $f=(1/2)f_r$ is especially important, because external timing wave superimposition at spectral null at $f=(1/2)f_r$ is reported to exhibit better characteristics than the usual self-timing in conjunction with ultra high speed PCM data transmission [23-24]. A simple algorithm to test whether or not the power spectrum vanishes at $f=(1/2)f_r$ using matrices will also be given.

3.2 Transmission Code with Spectral Null at $f=(1/2)f_r$

A transmission code is completely characterized by the state transition diagram of an encoding automaton. As an example, the state transition diagram of so-called bipolar code is shown in Fig.3.1. It completely represents the rules of bipolar code. That is to say, an input symbol 0 is represented by the absence of a pulse and an input symbol 1 by the existence of a pulse whose polarity is changed alternately. H.Yasuda and H.Inose have proven that the power spectrum of a transmission code has no *dc*-component if, and only if, it is generated

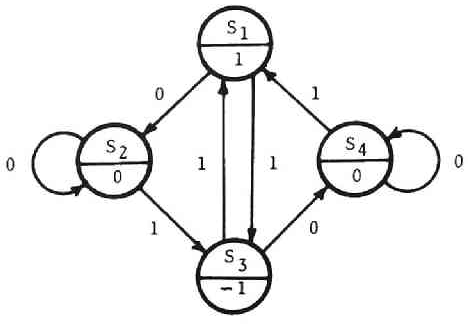


Fig.3.1 Encoding automaton of a bipolar code. (Output symbol associated with each state is shown in the circle.)

by a loop-sum-zero transition diagram. A transition diagram is called loop-sum-zero if, for any closed path in the state transition diagram of an encoder, the algebraic sum of the outputs along the path is equal to zero. Fig.3.1 shows an example of a loop-sum-zero transition diagram. Part of their results can be summarized as follows. Remember that power spectrum can be defined also by Eq.(2.13), which appeared in the preceding chapter.

[Proposition 3.1] [17-18] The following three statements are equivalent.

(i) $$w(0) = \lim_{\substack{N_1 \to -\infty \\ N_2 \to \infty}} \frac{|G(0)|^2}{(N_2 - N_1)T} \, \mathrm{av} \left| \sum_{i=N_1}^{N_2-1} a_i \right|^2 = 0 \ .$$

(ii) $$\forall_m, \forall_n \ (m<n), \quad \exists_M \quad s.t. \quad \left| \sum_{i=m}^{n} a_i \right| < M \ .$$

(iii) A state transition diagram of an encoding automaton is loop-sum-zero.

Starting from the above proposition, more general results were obtained. Namely, a necessary and sufficient condition for a transmission code to have a spectral null at frequency $f=(1/2)f_r$ is obtained. It is just as simple as Proposition 3.1 and may be useful when constructing a code with a spectral null at $f=(1/2)f_r$. The next Proposition 3.2 summarizes the result.

[Proposition 3.2] The following three statements are equivalent.

(i) $$w((1/2)f_r)=\lim_{\substack{N_1\to-\infty\\N_2\to\infty}}\frac{|G(\frac{1}{2}f_r)|^2}{2(N_2-N_1)T}\mathrm{av}\Big|\sum_{i=2N_1}^{2N_2-1}(-1)^i a_i\Big|^2=0.$$

(ii) $$\forall m,\ \forall n\ (m<n),\ \exists M\ \ s.t.\ \Big|\sum_{i=m}^{n}(-1)^i a_i\Big|<M.$$

(iii) Let $z_1, z_2, \ldots, z_{2m}$ $(m<N)$ be the outputs corresponding to any closed path of length $2m$ in an N-state transition diagram,

then $$\sum_{i=1}^{m} z_{2i-1}=\sum_{i=1}^{m} z_{2i}$$ always holds.

(Proof) Equivalence between statements (i) and (ii) in Proposition 3.2 is readily proven from the results of Proposition 3.1. Therfore, it is necessary only to prove the equivalence between statements (i) and (iii) in Proposition 3.2. If we let $f=(1/2)f_r$ in the power spectrum definition Eqs.(2.6),(2.12) and (2.13), we obtain

$$w((1/2)f_r)=\lim_{\substack{N_1\to-\infty\\N_2\to\infty}}\frac{|G(\frac{1}{2}f_r)|^2}{2(N_2-N_1)T}\mathrm{av}\Big|\sum_{i=2N_1}^{2N_2-1}(a_i-m)(-1)^i\Big|^2$$

$$= \lim_{\substack{N_1\to-\infty \\ N_2\to\infty}} \frac{|G(\frac{1}{2}f_r)|^2}{2(N_2-N_1)T} \text{ av}\Big|\sum_{i=2N_1}^{2N_2-1}(-1)^i a_i\Big|^2 .$$

Here, we define a new pulse sequence $\{b_k\}$, derived from $\{a_k\}$ by

$$b_k = a_{2k}-a_{2k+1} . \qquad (3.1)$$

Then, we get

$$w((1/2)f_r) = \lim_{\substack{N_1\to-\infty \\ N_2\to\infty}} \frac{|G(\frac{1}{2}f_r)|^2}{2(N_2-N_1)T} \text{ av}\Big|\sum_{i=N_1}^{2N_2-1} b_i\Big|^2 .$$

From Proposition 3.1, we can say that $w((1/2)f_r)=0$, if and only if $\{b_k\}$ is generated by a loop-sum-zero transition diagram. (As we are only interested in the characteristics independent of $G(f)$, it is assumed here that $G(f)\neq 0$ generally.) In order to check whether or not $\{b_k\}$ is produced by a loop-sum-zero transition diagram, it is necessary to test all the loops in the transition diagram of length up to N' (the number of states in the transition diagram of $\{b_k\}$). For N', we can let $N'=N$. This is due to the fact that it is possible to construct the transition diagram of $\{b_k\}$, whose state space is just the same as that of $\{a_k\}$. Thus, in order to check whether $w((1/2)f_r)=0$ or not, it is necessary to examine all the closed paths of length up to $2N$, to determine whether or not condition (iii) in Proposition 3.2 holds.

As an example, let us think of a transition diagram illustrated in Fig.3.2. Whether or not a tramsmission code, generated by the state transition diagram containing Fig.3.2 as its subgraph, has spectral null at $f=(1/2)f_r$ can be easily tested without carrying out direct calculation of the power spectrum. According to Proportion 3.2, it is necessary only to check the closed paths of even length and up to 16 in this case. In Fig.3.2, there are

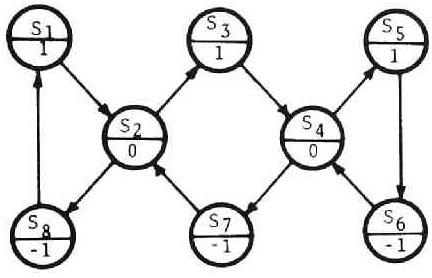


Fig. 3.2 Part of a state transition diagram.

several closed paths to be checked. For example, loops such as $s_1 \rightarrow s_2 \rightarrow s_8 \rightarrow s_1$, $s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_4$ and $s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_7 \rightarrow s_2$ do not violate the condition for $w((1/2)f_r)=0$, because the former two loops have odd lengths and the third one satisfies the required condition. The longer closed path $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_4 \rightarrow s_7 \rightarrow s_2 \rightarrow s_8 \rightarrow s_1$, however, does not meet the required condition, because the total summations of outputs taken over every other state along the path are not equal, i.e., +3 and -3, respectively. Thus, it can be concluded that the power spectrum does not vanish at $f=(1/2)f_r$ if a transmission code is generated by a transition diagram into which Fig.3.2 is embedded.

Now, it is shown that the testing algorithm for $w((1/2)f_r)=0$ described in the above, can be restated in terms of matrices which are more tractable and systematic. Let Q_1 and Q_2 be square matrices defined as follows:

$$Q_1 = (q^1_{ij}) \ , \quad Q_2 = (q^2_{ij}) \ , \tag{3.2}$$

where q^1_{ij} and q^2_{ij} are equal to zero if there is no direct state transition from s_i to s_j, and $q^1_{ij} = \sum_{k=1}^{l} z^{-a_k}$ and $q^2_{ij} = \sum_{k=1}^{l} z^{a_k}$ if there exist l input symbols which cause direct state transition from state s_i to s_j and the

associated outputs are a_k's. Furthermore, let R be matrix product Q_1Q_2. Then the following testing algorithm is derived.

[Corollary 3.1] A transmission code, whose transition diagram consists of N states, has spectral null at $f=(1/2)f_r$, if and only if there appears no term of z^i ($i\neq 0$) in the main diagonal of matrices R, R^2, ... , R^N.

If z^i ($i\neq 0$) exists in the diagonal of $R^m(1\leq m\leq N)$, it means that there is at least one closed path of length 2m such that the difference of total summations taken over every other state along the path is i, which contradicts the condition of $w((1/2)f_r)=0$.

As an example, let us take Fig.3.3 and apply the above corollary to test if $w((1/2)f_r)=0$. Matrices Q_1 and Q_2 and powers of R are calculated easily as follows.

$$Q_1=\begin{bmatrix} 0 & z^{-1} & 1 & 0 \\ 0 & z^{-1} & 1 & 0 \\ 1 & 0 & 0 & z \\ 1 & 0 & 0 & z \end{bmatrix}, \quad Q_2=\begin{bmatrix} 0 & z & 1 & 0 \\ 0 & z & 1 & 0 \\ 1 & 0 & 0 & z^{-1} \\ 1 & 0 & 0 & z^{-1} \end{bmatrix}, \quad R=\begin{bmatrix} 1 & 1 & z^{-1}z^{-1} & \\ 1 & 1 & z^{-1}z^{-1} & \\ z & z & 1 & 1 \\ z & z & 1 & 1 \end{bmatrix},$$

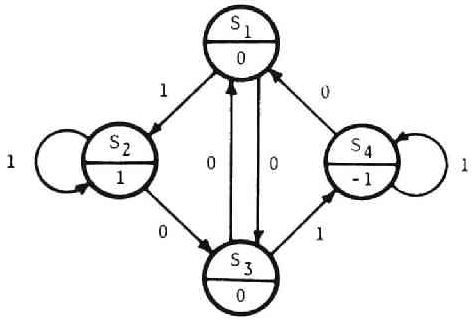
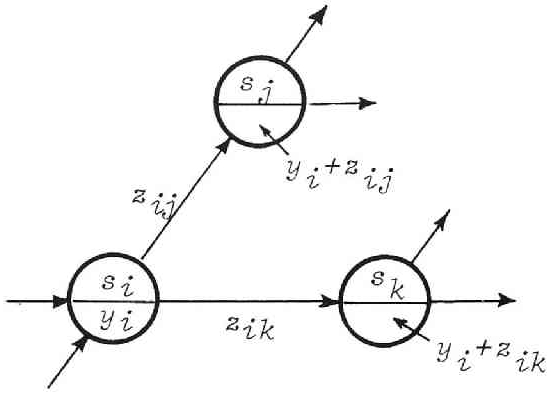


Fig. 3.3 Duobinary code state transition diagram.

$$R^2 = 4R \ , \quad R^3 = 4^2R \ , \quad R^4 = 4^3R.$$

In the main diagonal of R^i $(1 \leq i \leq 4)$, only integer 4^{i-1} appears. From these facts, it can be concluded that duobinary code shown in Fig.3.3 has a spectral null at $f=(1/2)f_r$ without direct calculation of power spectrum. If one derives a state transition diagram generating the sequence $\{b_k\}$ defined by (3.1), then a more elegant testing algorithm exists. It is briefly described here and will be utilized in the next chapter.

It is based on a notion of "state-level-number" assigned to each state, originally introduced by H.Yasuda and H. Inose [17]. This notion may be well recognized if one remembers that, in a static electromagnetic field, potential at any point relative to the reference point is uniquely determined. This is due to the fact that the integration of electric field strength $\mathbf{E}$ along the any loop path must be zero, i.e., $\oint_C \mathbf{E} \cdot d\mathbf{s} = 0$ (C:loop). This situation is analogous to the loop-sum-zero state transition diagram, where $\mathbf{E}$ corresponds to output symbol associated with each state transition and the integration of $\mathbf{E}$ along the loop corresponds to the summation of output symbols along the loop. It is the state-level-number that corresponds to the potential. That is, in a state transition diagram, if there exists a state transition from state s_i to s_j whose output symbol is z_{ij} , then the state-level-number of s_j is set to that of s_i plus z_{ij} (refer to Fig.3.4). If these assignments of state-level-number can be done without any contradiction for all the states and the associated state transitions, then we can conclude that the given diagram satisfies the loop-sum-zero condition. Conversely, if a loop-sum-zero state transition diagram is given, then we can assign the state-



z_{ij} : output symbol associated with a state transition from s_i to s_j.

Fig.3.4 Illustration of how to assign state-level-numbers (shown in the lower half of circles).

level-numbers to all the states without any contradiction.

Let us apply this procedure to a state transition diagram shown in Fig.3.3. First, we must derive the state transition diagram producing the output sequence $\{b_k\}$, i.e., expressing two step state transitions whose output sequence is $\{b_k\}$. It is shown in Fig.3.5(a). It should be noted that only the output symbol is assigned to each branch. Then, it is necessary to examine whether Fig.3.5 (a) is a loop-sum-zero state transition diagram or not. According to the above procedure, first assign a state-level-number, say 0, to some reference state, say s_1. As there exist four outgoing directed edges from s_1 in this case, assign to each destination state the state-level-number being equal to that of s_1 plus the associated output symbol. The result is shown in Fig.3.5(b). Following that, for any other state to which state-level-number is already assigned, repeat the same process to see if the assignment of the state-level-number coincides with preassigned state-level-number. In this case, the assignment shown in Fig.3.5(b) is compatible with all of the possible assignments. Thus we can conclude that the

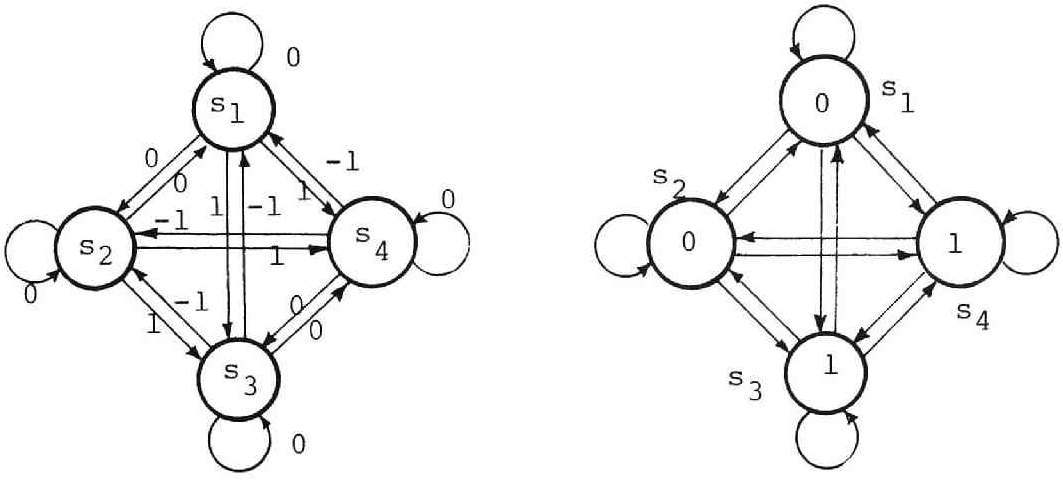


Fig.3.5 (a)State transition diagram derived from Fig.3.3 and (b) an example of assignment of state-level-numbers.

sequence generated by the encoding automaton of Fig.3.3 has a spectral null at $f=(1/2)f_r$.

3.3 Transmission Code with Spectral Null at $f=(h/k)f_r$

Let us consider a more general case. Namely, we shall examine the condition for a transmission code to have a spectral null at frequency $f=(h/k)f_r$, where h and k are nonnegative and positive integers, respectively. If we let $f=(h/k)f_r$ in the definition of power spectrum Eqs.(2.6),(2.12) and (2.13), we have

$$w((h/k)f_r)=\lim_{\substack{N_1\to-\infty\\N_2\to\infty}}\frac{|G(h/kf_r)|^2}{k(N_2-N_1)T}\,\mathrm{av}\Big|\sum_{i=kN_1}^{kN_2-1}a_i e^{-j2\pi i(h/k)}\Big|^2. \quad (3.3)$$

If we define a new sequence $\{c_k\}$ by the following relation

$$c_n = \sum_{i=0}^{k-1} a_{kn+i} e^{-j2\pi i(h/k)} \quad , \tag{3.4}$$

Eq.(3.3) can be rewritten as follows:

$$w\ ((h/k)f_r) = \lim_{\substack{N_1\to-\infty \\ N_2\to\infty}} \frac{|G(h/k f_r)|^2}{k(N_2-N_1)T} \mathrm{av} \left| \sum_{i=N_1}^{N_2-1} c_i \right|^2 . \tag{3.5}$$

Then, from Proposition 3.1, it is concluded that $w((h/k)f_r)=0$, if and only if $\{c_k\}$ is generated by a loop-sum-zero transition diagram. To determine whether $\{c_k\}$ is produced by such a transition diagram, it is necessary to examine the closed paths of length up to kN and multiple of k in the transition diagram of $\{a_k\}$. This is due to the fact that it is possible to construct a transition diagram of $\{c_k\}$ whose state space is just the same as that of $\{a_k\}$. Thus, from the above discussion and relation (3.4), the next useful proposition can be derived.

[Proposition 3.3] The following three statements are equivalent.

(i) $w((h/k)f_r)= 0$.

(ii) $\forall_m, \forall_n \quad (m<n), \exists_M \quad s.t. \quad \left| \sum_{i=m}^{n} a_i e^{-j2\pi i(h/k)} \right| < M$

(iii) Let $z_1, z_2, \ldots, z_{km}$ $(m<N)$ be the outputs corresponding to any closed path of length km in an N-state transition diagram. Then

$$\sum_{i=1}^{km} z_i e^{-j2\pi i(h/k)} = 0 \ . \tag{3.6}$$

This proposition would be the most general result in

the problem of power spectral nulls, where h/k = 0/1 corresponds to the case of dc-component free code and h/k = 1/2 corresponds to the case discussed in the preceding section. If we let h/k = 1/3, the condition of $w((1/3)f_r)=0$ is readily obtained, such that three total summations taken over every three states along the closed path of length $3m$ ($m \leq N$) coincide with each other. From the above proposition, it can be said that the concept of loop-sum-zero transition diagram, in the case of a dc-component free code, has been generalized to the complex-value weighted loop-sum-zero condition (corresponding to Eq.(3.6)) with respect to the specific closed paths of length up to kN and multiple of k. Note that all propositions stated in this chapter hold true for any state transition diagram including that having periodic structure.

CHAPTER 4 CATALOG OF MEALY-TYPE ENCODING AUTOMATA WITH SPECTRAL NULLS

Based on the results in Chapter 3, a catalog of encoding automata producing the pulse sequences with power spectral nulls at f=0 or $(1/2)f_r$ is given in this chapter. This catalog exhaustively lists all possible Mealy-type encoding automata with up to 4 states under the condition that input symbol 0 always produces output symbol 0 and input symbol 1 produces output symbol either 1 or -1. Totally, a single 2-state automaton and nine 4-state automata with a spectral null at f=0, and a single 2-state automaton, two 3-state automata, and thirteen 4-state automata with a spectral null at $f=(1/2)f_r$ are obtained.

4.1 INTRODUCTION

Although we have discussed some features of transmission codes whose output sequences have power spectral nulls at $f=(h/k)f_r$, we have not yet presented how to construct such a transmission code. Development of a general method for such a problem is of great significance, but is surely not an easy task. However, if restricted to a special class of transmission codes, actual construction of them may be possible. In this chapter, by restricting the input/output symbols assignment to 0/0 and

either 1/1 or 1/-1, we will construct a catalog of so called pseudoternary transmission codes with power spectral nulls at f=0 or (1/2)f_r.

Since the necessary and sufficient condition for transmission code sequences to have power spectral nulls at f=0 or (1/2)f_r is given in terms of closed paths in state transition diagrams, we will first construct digraphs (directed graphs) satisfying the required condition and then assign input/output symbols so as to meet the necessary and sufficient condition.

As a first step, taking the necessary conditions for a given digraph to become an encoding automaton by proper input/output symbols assignment into account, we have made up a list of different strongly connected digraphs with up to 4 nodes and outdegree 2. Totally 3 such digraphs with 2 nodes, 14 with 3 nodes and 108 with 4 nodes are obtained up to isomorphism.

Also, we have examined these digraphs in terms of eigenvalues of its adjacency matrix with the result that a strongly connected digraph with no more than 3 nodes and outdegree 2 can be uniquely identified by the eigenvalues up to isomorphism. However, in the case of 4 nodes, it does no longer hold true, yet has proven to be useful in clustering the set of digraphs roughly up to isomorphism.

On the basis of these digraphs, we tried to construct encoding automata by assigning input and output symbols to each edge of these digraphs. As a result, 1) a single encoding automaton with 2 states, 9 encoding automata with 4 states with power spectral null at f=0; 2) a single encoding automaton with 2 states, 2 encoding automata with 3 states, 13 encoding automata with 4 states with power spectral null at f=(1/2)f_r; 3) a single encoding automaton with 4 states with power spectral nulls at both f=0 and

$(1/2)f_r$, are listed in our catalog. The last one is also included in both 1) and 2).

Moreover, we have shown a method to make an encoding automaton with more than four states having power spectral null at f=0 (or f=$(1/2)f_r$) by combining two encoding automata with power spectral null at the same frequency.

The catalog generated in this chapter is believed helpful in selecting a transmission code with a power spectral null at f=0 or $(1/2)f_r$ or at both.

4.2 Preliminaries for Catalog Construction

In this chapter, we deal with only a Mealy-type encoding automaton, which is generally defined as quintuple $M = (S,X,Y,\delta,\lambda)$, where S is a finite set of states, X=$\{0,1\}$ is a set of input symbols, Y=$\{-1,0,1\}$ is a set of output symbols, δ:$S\times X\rightarrow S$ is a next state function and λ: $S\times X\rightarrow Y$ is an output function. That is, we focus upon a class of codes so called pseudoternary transmission codes [1]. In addition, we restrict the set of combinations of input/output symbols to only 0/0 and either 1/1 or 1/-1. This restriction imposed on input/output relation is so strong that one may suppose that we might overlook valuable encoding automata. However, most of the widely used and important transmission codes from the practical point of view are in fact in this class.

Before proceeding to the construction of a catalog, we will carry out enumeration of finite automata based on

the formula of Harary et al.[25] to know the approximate number of automata with which we are concerned. Table 4.1 shows the number of different finite automata with binary input and ternary output up to isomorphism. It shows there exist more than a million automata even in the case of four states. As an encoding automaton, however, the strongly connected finite automaton is preferable from the practical point of view. Consequently, we are going to search for the strongly connected encoding automata with specified property.

Table 4.1 Number of finite automata with binary input and ternary output symbols (up to isomorphism).

Number of States	Number of Finite Automata
2	74
3	7,623
4	1,501,516
5	401,371,270

4.3 Catalog of Strongly Connected Digraphs

In order to list all the strongly connected digraphs systematically, we have used a vector representation of a digraph instead of an adjacency matrix representation[26]. That is to say, we assigned a unique number to each node

of a digraph according to what nodes the two outgoing edges reach relative to the starting node. It corresponds to assign a unique number to each row of an adjacency matrix , taking the position of the row into account.

For instance, for a 3-node digraph we have assigned the number to each node as shown in Fig.4.1. Namely, any digraph can be represented by 3-tuple (x_1, x_2, x_3) where $x_i \in \{0, \pm1, \pm2, 3\}$ $(i=1,2,3)$.

As there are isomorphic digraphs and non-strongly connected digraphs among the candidate digraphs thus obtained, we must locate and exclude them. By using the vector representation we can check the isomorphism and strong connectivity as follows in the case of 3-node digraphs. That is, for an arbitrary digraph with the representation (x_1, x_2, x_3), the digraph which has either

(1) the vector representation obtained by a cyclic shift of (x_1, x_2, x_3)

or

(2) the representation $(x_1, -x_3, -x_2)$

is isomorphic.

Also we can say that the digraph which has either

(a) the representation ((-2 or -1),(2 or 1),*) or (3,*,*)

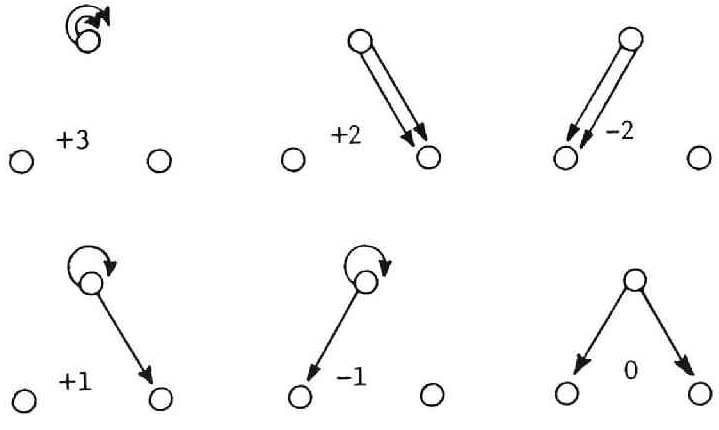


Fig. 4.1 Assignment of a unique number to each node based on configuration of outgoing edges.

or

(b) the representation of the digraph which is isomorphic to the digraph of (a)

is not strongly connected, where * means arbitrary number.

From these considerations, we have obtained the objective strongly connected digraphs. Namely, 14 strongly connected digraphs with 3 nodes and 108 strongly connected digraphs with 4 nodes up to isomorphism. All the strongly connected digraphs with 3 nodes are shown in Fig.4.2.

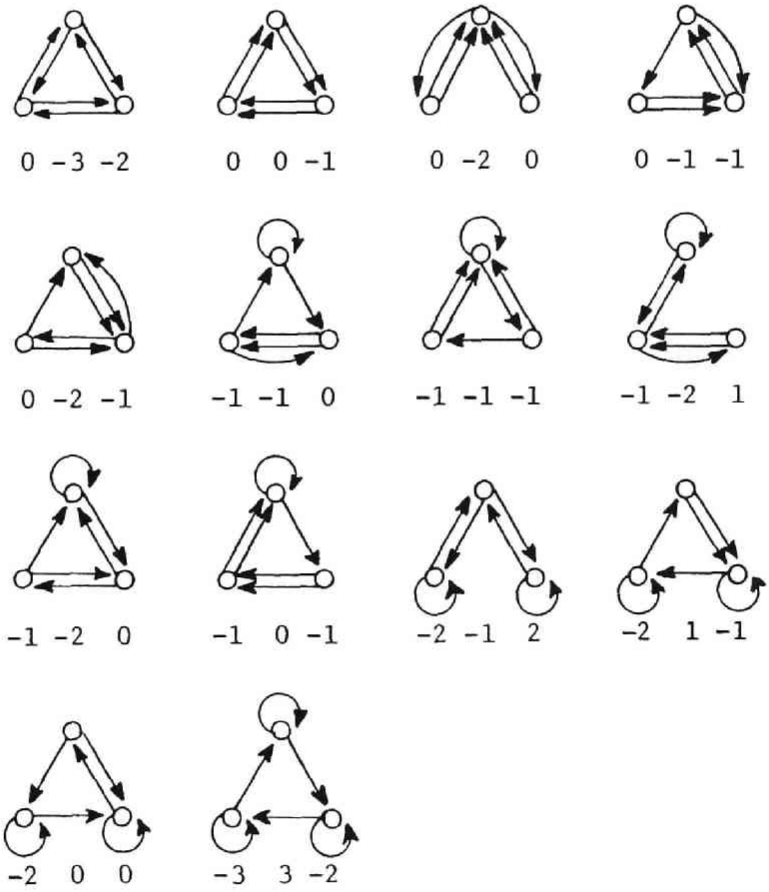


Fig. 4.2 A list of strongly connected digraphs with 3 nodes and outdegree 2, and the coefficients (a_0, a_1, a_2) of characteristic polynomials of $x^3+a_0x^2+a_1x+a_2$ of their adjacency matrices.

Here, we shall describe a characterization of a digraph by the eigenvalues of its adjacency matrix which has proven to be helpful for confirming non-isomorphism of digraphs. The adjacency matrix $A=(a_{ij})$ of a digraph is defined as follows:

$$a_{ij} = \begin{cases} 1 \ldots \text{if there exists an edge from the } i\text{-th node to the } j\text{-th node,} \\ 0 \ldots \text{otherwise.} \end{cases}$$

When two digraphs are given and the eigenvalues of their adjacency matrices are completely equal, the two digraphs are called cospectral [27]. It is known that cospectrality is necessary but not sufficient for two digraphs to be isomorphic. Eigenvalues are, as is well-known, the roots of the characteristic equation. Therefore, by checking the coefficients of the characteristic polynomial, we can decide whether two digraphs are non-isomorphic.

Since we are considering the only special class of digraphs which consists of strongly connected digraphs with outdegree 2, there may be one to one mapping between digraphs and eigenvalues of their adjacency matrices up to isomorphism. This conjecture is true with respect to strongly connected digraphs with 3 nodes as shown in Fig. 4.2, but not true with respect to those with 4 nodes. That is, it is shown that a strongly connected digraph with 3 nodes having outdegree 2 is uniquely identified by the eigenvalues of its adjacency matrix. However, as for strongly connected digraphs with 4 nodes, it does not hold true, yet has proven to be very useful for rough classification of digraphs up to isomorphism. In fact, of 108 non-isomorphic strongly connected digraphs with 4 nodes, there exist a pentad of cospectral non-isomorphic digraphs, shown in Fig.4.3, as well as 2 tetrads, 4 triads and so on. The result of classification of non-isomorphic digraphs based on cospectrality is summarized in Table 4.2.

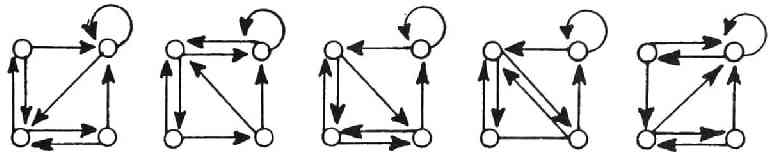


Fig. 4.3 Five non-isomorphic digraphs with the same characteristic polynomials $x^4-x^3-2x^2$.

Table 4.2 Classification of 108 strongly connected digraphs with 4 nodes by means of cospectrality.

n	1	2	3	4	5
Number of Classes of n Cospectral Digraphs	51	16	4	2	1

4.4 Catalog of Encoding Automata and Their Power Spectra

By assigning input/output pairs of symbols to all the edges of strongly connected digraphs, we can obtain a catalog of encoding automata. It should be also examined if the resultant automata are state minimized or not.

In order to carry out the assignment of input/output pairs of symbols efficiently, we listed some necessary conditions with regard to subgraphs. They are depicted in Fig.4.4. The second one in Fig.4.4 (1), for instance, says that if there exist two edges directly connecting two states, then they must be provided with the same out-

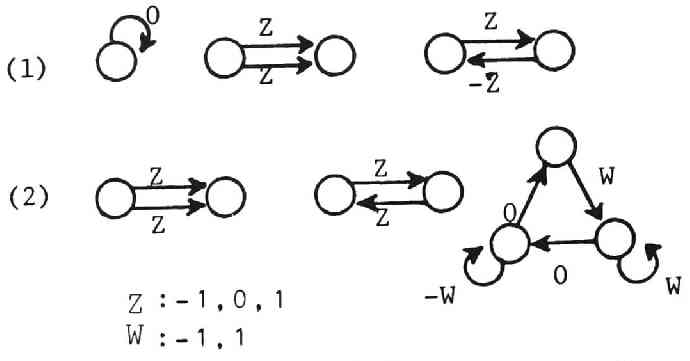


Fig. 4.4 Necessary conditions concerning output symbols assignment to the subgraphs ((1) for the case of spectral null at f=0 and (2) for the case of spectral null at $f=(1/2)f_r$).

put symbols. However, from the assumption that two edges originating the same state must be assigned the different output symbols, the graphs in which it is embedded must be excluded from the candidates of catalog.

Thus, if we delete strongly connected digraphs having a node with two edges reaching the same state, then we have

1 digraph with 2 nodes,
5 digraphs with 3 nodes,
33 digraphs with 4 nodes,

as candidates.

As the total number of candidate digraphs is relatively decreased, we will pursue the encoding automaton by assigning input/output symbols in a cut and try fashion, with the necessary conditions in Fig.4.4 in mind.

Once an assignment is done, the verification whether it satisfies the condition of spectral null or not can be immediately carried out by the method based on the state-level-number as discussed in the preceding chapter. This is executed easily by digital computer. However, hand calculation is also possible in this case.

Using a digital computer PDP 11/40, we have derived all the minimal encoding automata which satisfy the required conditions. They are all listed in Fig.4.5, and, at the same time, the associated normalized power spectra are listed in Fig.4.6. They are classified according to the number of states and the frequencies of spectral nulls. The calculation of power spectra is carried out by the matrix-based method, mentioned in Chapter 2, assuming the equal occurrence probabilities of binary input symbols. Although we have tried to find an encoding automaton with spectral null at $f=(1/3)f_r$ (or $(2/3)f_r$), we could not find it with up to four states. Some comments on encoding automata in a catalog are as follows.

(1) *Encoding automata generating the sequences with spectral null at f=0 (dc).*

A two-state encoding automaton in a catalog corresponds to the well-known bipolar code. There exists no three-state encoding automaton in it. Totally nine encoding automata with four states exist. Among them, each pair of automata M_2 and M_2^*, M_4 and M_4^*, and M_5 and M_5^* are isomorphic as digraphs and the corresponding power spectra are the same, because the output sequence of one automaton is the negative of the other with its sign in each pair. The M_6 is known as a count-2 bipolar code.

(2) *Encoding automata generating the sequences with spectral null at $f=(1/2)f_r$.*

A two-state encoding automaton corresponds to the well-known duobinary code. There exist two three-state and thirteen four-state encoding automata. Among them, each pair of automata M_7 and M_7^*, M_9 and M_9^*, M_{10} and M_{10}^*, M_{11} and M_{11}^*, M_{12} and M_{12}^* and M_{13}

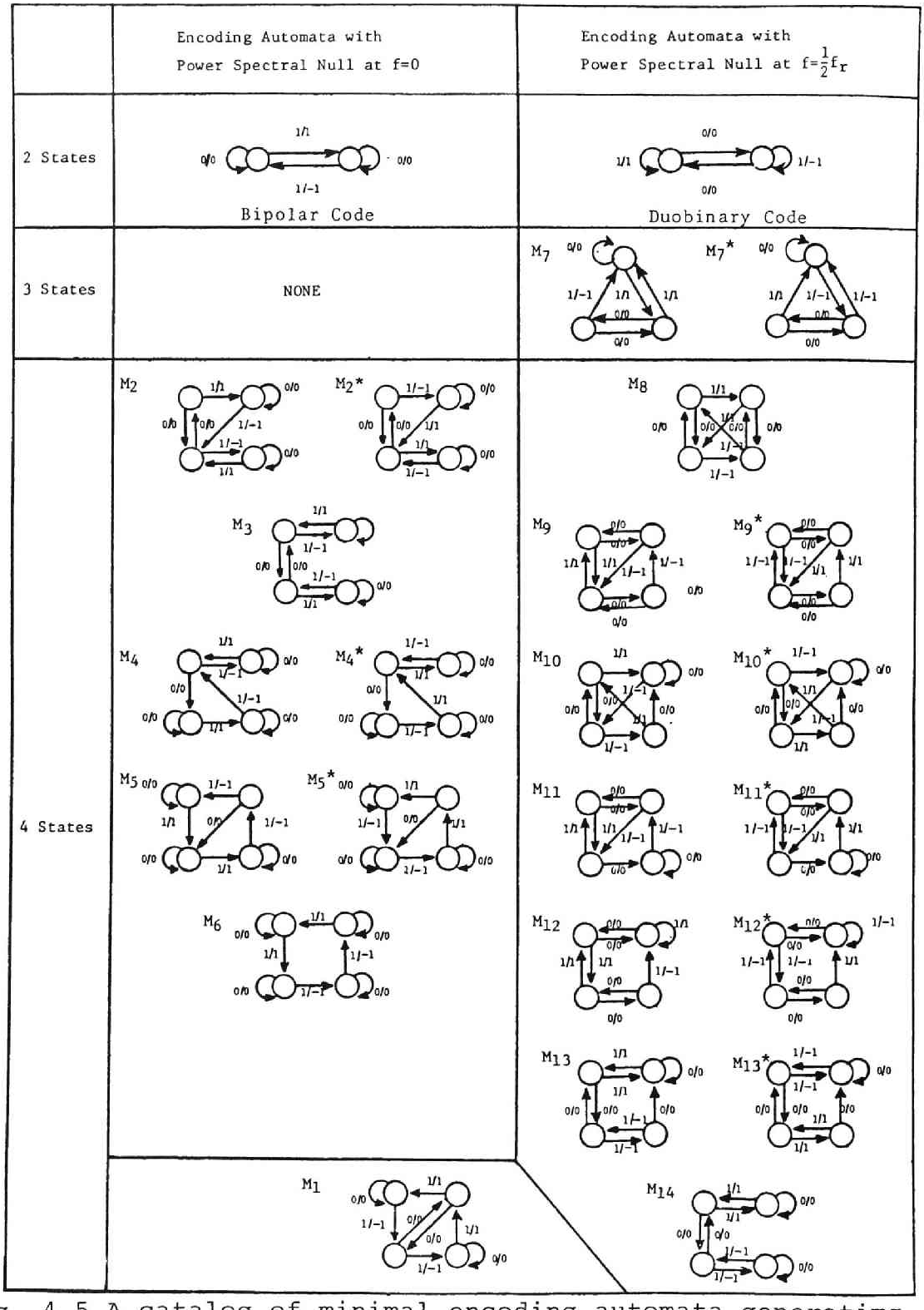


Fig. 4.5 A catalog of minimal encoding automata generatimg sequences with power spectral null at f=0 or (1/2)f_r or at both.

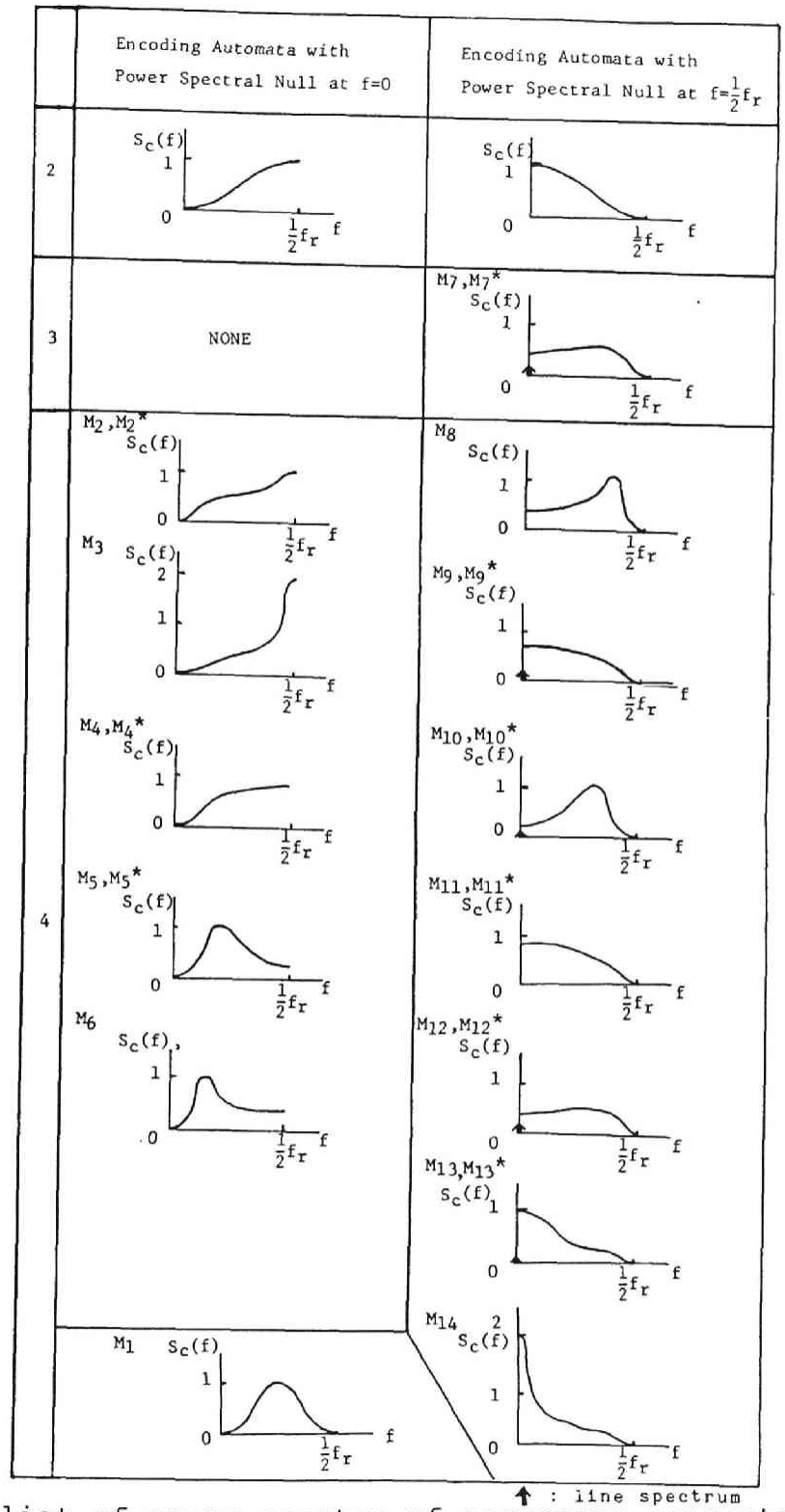


Fig. 4.6 A list of power spectra of sequences generated by the corresponding encoding automata shown in Fig.4.5.

and M_{13}^{*} are isomorphic as digraphs and the corresponding power spectra are the same in each pair.

Only a single encoding automaton whose output sequence having spectral nulls both at f=0 and $(1/2)f_r$ is obtained.

Let us now consider some heuristic method for constructing encoding automata with more than four states from already obtained automata with fewer states.

First, a method which takes advantage of a self loop is presented, when an encoding automaton with spectral null at f=0 is given. It constructs a new encoding automaton , with number of states increased by one , having the same spectral null. The major steps of the procedure are summarized as follows.

Step 1 : Find a state with a self-loop to which 0/0 is assigned and let it be s_i.

Step 2 : Add a single new state s_i' and replace a self-loop of s_i by a state transition from s_i to s_i' to which 0/0 is assigned.

Step 3 : Find a state s_j ($j \neq i$) such that outputs summed along the path from s_i to s_j equals to +1 or -1, which we denote z_{ij}. Then introduce a state transition from s_i' to s_j to which $1/z_{ij}$ (input/output symbols) is assigned.

Step 4 : Find a state s_k such that outputs summed along the path from s_i to s_k equals to zero. Then, introduce a state transition from s_i' to s_k to which 0/0 is assigned.

Step 5 : Make a state minimization if possible.

By applying the above procedure once or successively if possible, we can get various encoding automata with more than four states. Two examples of them together with

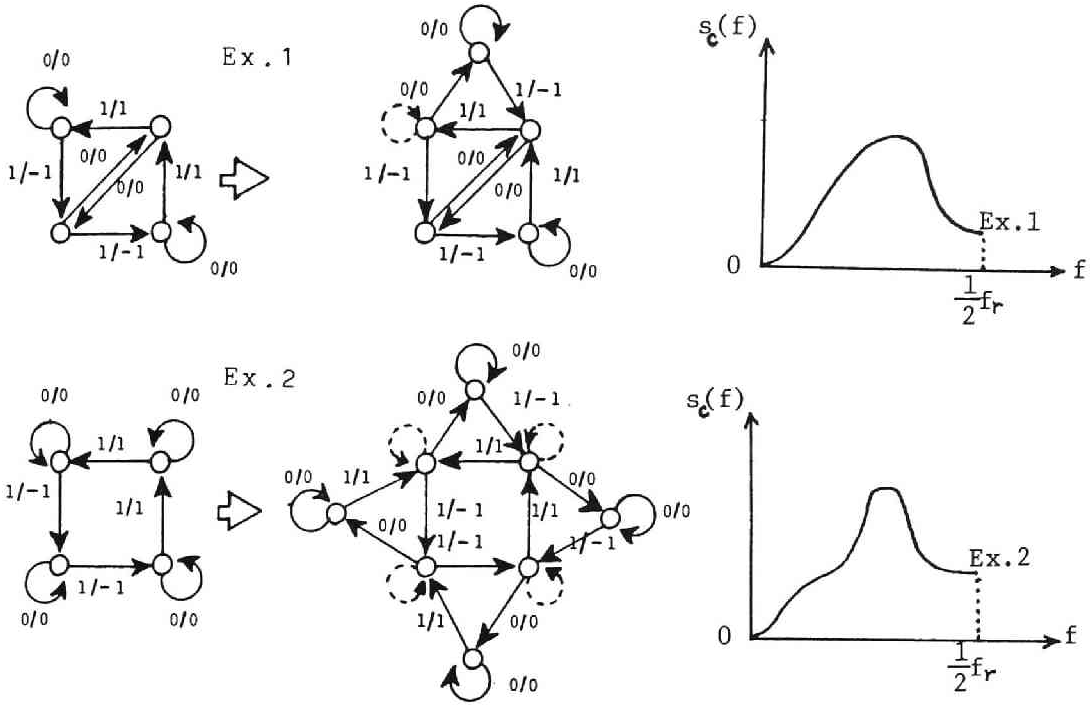


Fig. 4.7 Examples of encoding automata with spectral null at f=0 constructed by modifying state transition diagrams, and their power spectra.

their power spectra are shown in Fig.4.7.

The next procedure to make larger encoding automata is applicable to both cases of having spectral nulls at f=0 or $(1/2)f_r$.

[Proposition 4.1] Let two automata whose output sequences have spectral nulls at f=0 (or $(1/2)f_r$) be M_A and M_B. Moreover, both of them are assumed to have at least one state with a self-loop whose output symbol is the same. Then a new encoding automaton is obtained by modifying the state transitions of those states with self-loops, i.e., by replacing the self-loops by state transitions leading to the other state with a self-loop as

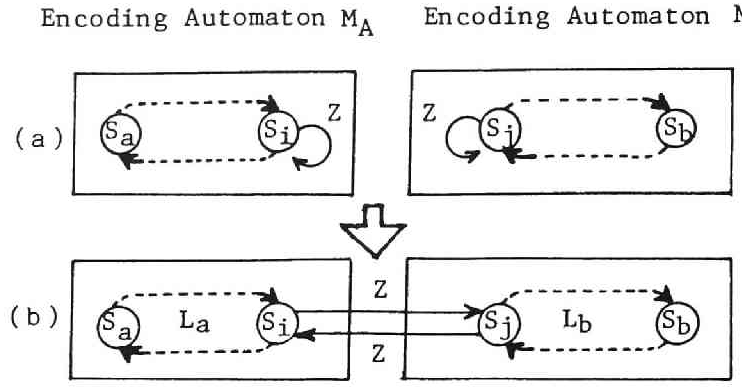


Fig.4.8 Illustration of the process of combining two automata to make a larger one with spectral null at either $f=0$ or $(1/2)f_r$.

illustrated in Fig.4.8.

(proof) We shall prove the proposition only for the case of power spectral null at $f=(1/2)f_r$, since almost the same procedure is applicable to the case of $f=0$. If a newly generated closed path of even length exists, it must be of the type shown in Fig.4.8(b), where both closed paths in the original automata are either of even length or odd length. Let us assume that both of them are of odd length, and let the output sequences along L_a and L_b starting from s_i and s_j, respectively, be $z_1, z_2, \cdots, z_{2l+1}$ and $z'_1, z'_2, \cdots, z'_{2m+1}$. Then, in a closed path $s_i \to s_a \to s_i \to s_j \to s_b \to s_j \to s_i$, the summations taken every other output symbols are $(z_1+z_3+\ldots+z_{2l+1})+(z'_1+z'_3+\ldots+z'_{2m+1})$ and $(z_2+z_4+\ldots+z_{2l})+z+(z'_2+z'_4+\ldots+z'_{2m})+z$. On the other hand, from the condition that both original automata have spectral nulls at $f=(1/2)f_r$, the relations $(z_1+z_3+\ldots+z_{2l+1})=z+(z_2+z_4+\ldots+z_{2l})$ and $(z'_1+z'_3+\ldots+z'_{2m+1})=z+(z'_2+z'_4+\ldots+z'_{2m})$ hold true. Thus, the new closed path meets the condition that the composite encoding automaton produces a spectral null

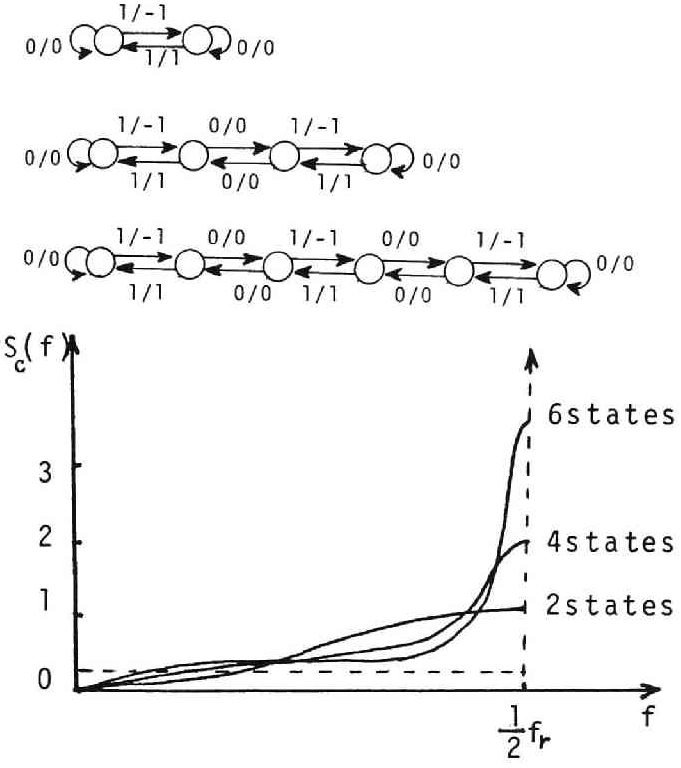


Fig.4.9

Encoding Automata constructed by a sort of concatenation of bipolar encoding automata and their power spectra.

(The dotted line shows power spectrum of TPC code, which is obtained by infinite concatenation.)

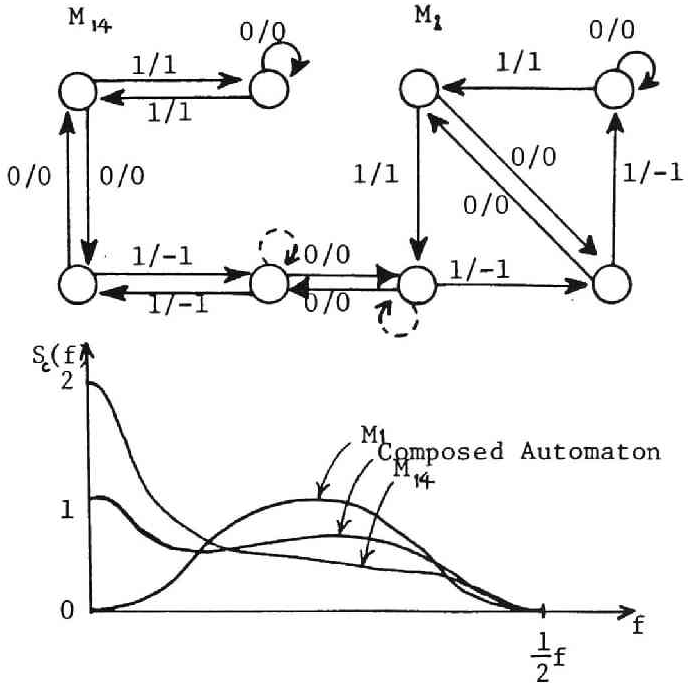


Fig.4.10

An encoding automaton constructed by combining M_1 and M_{14} (shown in Fig.4.5) and the associated power spectra.

at $f=(1/2)f_r$.

By applying the above procedure, we can make a variety of new encoding automata. Some of them are shown in Fig.4.9 and Fig.4.10. Fig.4.9 illustrates the procedure for obtaining the encoding automata with arbitrary even number of states by a sort of concatenation of bipolar encoding automata. It is of interest to consider the limiting behavior of power spectrum as the number of states approaches infinity. Closer inspection of limiting behavior of state transition diagrams reveals the fact that it is TPC (time polarity control) code that is obtained by infinitely applying the above procedure. It is to be noted that TPC code does no longer satisfy the requirement of *dc*-component free and it has line spectrum at $f=(1/2)f_r$. Fig.4.10 shows the encoding automaton constructed from M_1 and M_{14}. Other examples of heuristic construction of larger encoding automata are described in [28].....

With respect to an encoding automaton with a spectral null at $f=(1/3)f_r$, it is proven that there is no such automata with up to four states. However, we can give an example with eight states based on the theory of partial response code. It is shown in Fig.4.11. It is interesting to know what is the minimum number of states of such encoding automata.

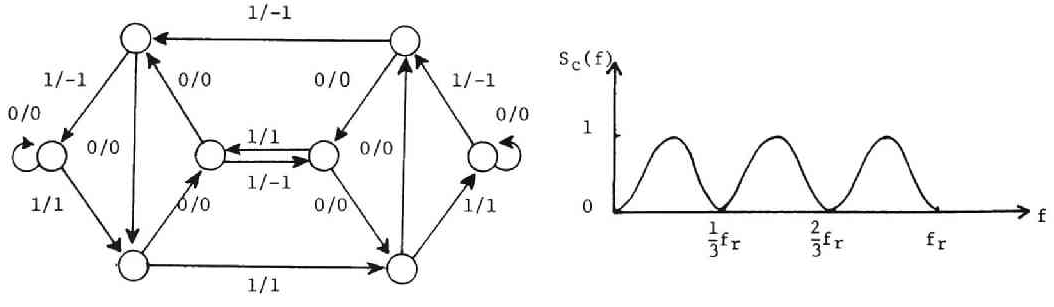


Fig.4.11 An encoding automaton with 8 states generating the sequence with spectral null at $f=(1/3)f_r$.

CHAPTER 5 RESHAPING OF POWER SPECTRA OF DIGITAL CODED SIGNALS BY RESTRICTING INPUTS TO ENCODING AUTOMATA

In this chapter, reshaping of power spectra of digital coded signals, caused by input restriction to an encoder, is dealt with. Especially, we will show that $(h/k)f_r$ spectral component can be removed for some transmission codes by properly restricting the input sequences. It is also shown that line spectra can be generated by such input restriction.

5.1 Introduction

When we discuss about spectral characteristics of a transmission code, it is usually assumed that input sequences are independent time series with predetermined occurrence probabilities of symbols 0 and 1. It is this independency that the mutual information can be maximized in a symmetric channel. However, input sequences are sometimes not independent but rather correlated because of some constraints imposed on them. Those rules which are employed to format the sequences for error control or for data link control are typical examples of such constraints. When an input sequence with correlated statistics is applied to the encoder, the power spectrum of the output sequence will be naturally different from

that when an independent input sequence is applied. Consequently, if the inputs to the encoder are properly restricted to its proper subset, which implies the introduction of some kind of correlation into inputs, more desirable spectral characteristics may be obtained without reconstructing the encoder.

As spectral reshapings, we will think of the elimination of specific frequency component, i.e., placement of nulls in the power spectrum, and generation of discrete line spectrum. Firstly, it is shown that a transmission code with *dc*-component can be made *dc*-component free by only restricting inputs, by selecting *DSV* (digital sum variation) [1] as the parameter of *dc*-component elimination. In addition, discussion is made as to what extent it is necessary to reduce the transmission rate in order to realize such a spectral transformation.

Then, extending the discussion, we will deal with the case of $(1/2)f_r$-component elimination or more generally $(h/k)f_r$-component elimination of transmission codes by properly restricting the set of input sequences. By defining the weighted *DSV* at the specific frequency as the new parameter, we will show how to calculate the limiting transmission efficiency, to be defined later.

Lastly in this chapter, we will discuss the possibility that transmission code sequences will produce line spectra when the set of input sequences is restricted properly.

5.2 Generation of Spectral Nulls

5.2.1 *dc*-Component Elimination

A transmission code is *dc*-component free if the total sum of the outputs of any time duration can be finitely bounded. Among such codes, *ac*-coupling distortion suffered by a coded signal is proportional to the digital sum (*DS*), where *DS* at time *K* is defined by $\sum_{n=0}^{K} a_n$, and digital sum variation (*DSV*), defined by the difference between the upper and lower bounds of *DS*, limits the distortion of any coded signal according to A.Croisier [1]. Therefore, an investigation was made on the limiting transmission rate or efficiency of a restricted code, assuming that *DSV* is given.

Let $N(n)$ and $N_R(n)$ be the numbers of possible input sequences without and with *DSV* restriction, respectively. Then, the limiting transmission efficiency of a restricted code is defined as follows:

$$\eta = \lim_{n\to\infty} \frac{\log_2 N_R(n)}{\log_2 N(n)} = \lim_{n\to\infty} \frac{\log_2 N_R(n)}{n} , \quad (5.1)$$

assuming that the input symbol is binary. If η is not equal to zero, data transmission without *dc*-component can be realized theoretically by taking only $1/\eta$ times longer time, compared with that of without input restriction. Main objectives are to find an efficient η calculation method and to search for a practical coding method to realize the limiting transmission efficiency η. The former question is considered first.

In order to calculate $N_R(n)$ numerically, it is neces-

sary to enumerate the number of possible input sequences of length n whose DS's do not exceed the given bound. This is done by the following procedure. A composite automaton is made whose state space is a direct product $S \times D$, where S is the state space of an original encoding automaton and D is the set of $(d+1)$ consecutive integers $\{[(1-d)/2], \ldots, -1, 0, 1, \ldots, [(1+d)/2]\}$ assuming that $DSV=d$ is given, where $[x]$ denotes the largest integer not exceeding x. Formally speaking, a composite automaton

$$M_c = (S \times D, X, Y, \delta_c, \lambda_c)$$

is constructed from the original encoding automaton $M = (S, X, Y, \delta, \lambda)$ by defining the next state function δ_c and the output function λ_c as follows, i.e., for any $(s,d) \in S \times D$ and any $x \in X$

$$\delta_c((s,d),x) = \begin{cases} (\delta(s,x), d+\lambda(s,x)) & \text{if } d+\lambda(s,x) \in D \\ \text{undefined} & \text{if } d+\lambda(s,x) \notin D, \end{cases}$$

$$\lambda_c((s,d),x) = \begin{cases} \lambda(s,x) & \text{if } d+\lambda(s,x) \in D \\ \text{undefined} & \text{if } d+\lambda(s,x) \notin D, \end{cases}$$

where $X=\{0,1\}$ and Y are sets of the input symbols and output symbols, respectively.

Since interest is in a stationary coded signal, the transient states are excluded from the composite finite automaton. A state transition matrix (adjacency matrix) A for the composite automaton is defined according to the following rule, i.e., its (i,j) entry is equal to 1 if a direct state transition from i to j exists and 0 otherwise. Then, the number of possible allowed paths of length n in the state transition diagram constructed above is calculated as

$$N_R(n) = (0\ldots\underset{\underset{i}{\smile}}{0}1\ldots0)\ A^n \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \qquad (5.2)$$

assuming that the state i is the initial state. It should be noted that η does not depend upon the initial state, although the proof is not given here. Thus, in order to evaluate $N_R(n)$, it is necessary to calculate the powers of matrix A. However, it is not necessary if the objective is only calculation of η. Using Frobenius' theorem on non-negative matrix [29], the following useful proposition can be derived.

[Proposition 5.1] The limiting transmission efficiency η is equal to $\log_2 \lambda_{max}$, where λ_{max} is the largest positive eigenvalue of the state transition matrix of composite automaton defined in the above.

(Proof) It follows, from the Frobenius' theorem, that the irreducible non-negative matrix, say A, has a positive eigenvalue λ_{max}, which is greater than or equal to the magnitude of all other eigenvalues. Furthermore, the eigenvectors corresponding to λ_{max} consist of all multiples of some vector $\mathbf{X}$ whose components x_i $(i=1,2,\ldots)$ are all positive. There exist some small positive number ε and some large positive number M, such that

$$\varepsilon \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} < \mathbf{X} < M \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} .$$

Multiplying $(0\ldots0\underset{i}{1}0\ldots0)A^n$ to each expression of the above inequality from the left , we obtain

$$\varepsilon(0\ldots0\underset{i}{1}0\ldots0)A^n \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} < \lambda_{max}^n x_i < M(0\ldots0\underset{i}{1}0\ldots0)A^n \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} ,$$

which, in turn, can be rewritten using Eq.(5.2) as

$$\frac{x_i}{M}\ \lambda_{max}^{n} \ < \ N_R(n) \ < \ \frac{x_i}{\varepsilon}\ \lambda_{max}^{n}$$

Thus, it is possible to obtain the next inequality

$$\log_2\lambda_{max}+\frac{1}{n}\log_2\left(\frac{x_i}{M}\right) \ < \ \frac{\log_2 N_R(n)}{n} \ < \ \log_2\lambda_{max}+\frac{1}{n}\log_2\left(\frac{x_i}{\varepsilon}\right),$$

which implies, as $n\to\infty$,

$$\eta = \lim_{n\to\infty}\frac{\log_2 N_R(n)}{n} = \log_2\lambda_{max} \ .$$

As an illustrative example, we consider two well-known codes, i.e., duobinary code and TPC (time polarity control) code, whose Mealy-type state transition diagrams are shown in Fig.5.1. In particular , how to calculate η for TPC code is shown under the condition of DSV=2. The state transition diagram of the composite finite automaton is shown in Fig.5.2. The state transition matrix A, associated with the automaton is readily determined from Fig.5.2. The largest positive eigenvalue of A is obtained

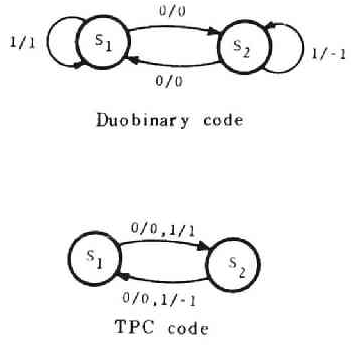


Fig.5.1 State transition diagrams for duobinary and TPC codes.

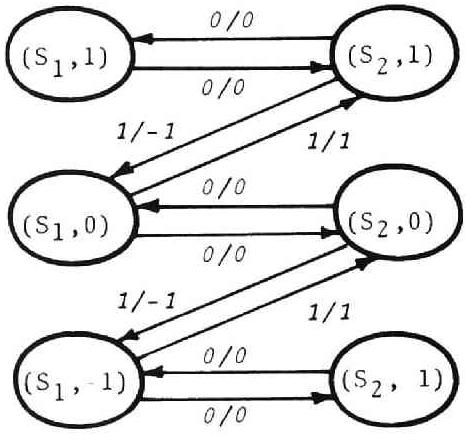


Fig.5.2 State transition diagram for TPC code with DSV=2.

$$A = \begin{array}{c} \\ (s_1,\ 1) \\ (s_1,\ 0) \\ (s_1,-1) \\ (s_2,\ 1) \\ (s_2,\ 0) \\ (s_2,-1) \end{array} \left(\begin{array}{ccc:ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right)$$

by solving the characteristic equation of A. In this case, λ_{max} is 1.802. Therefore $\eta = \log_2 \lambda_{max} = 0.850$. By the same procedure as the above, η for duobinary and TPC codes has been calculated under the condition that *DSV* varies from 1 to 4. The results are given in Table 5.1. According to the Table 5.1, we see that η for duobinary code of *DSV*=1 is 0.5. This limiting transmission efficiency η is easily achieved by simply inserting redundant input symbol 0 after every source input symbol. Namely, assuming that the input sequence is ..., a_{-2}, a_{-1}, a_0, a_1, a_2, ..., it can be transformed into *dc*-component free code by only changing it into ..., a_{-2}, 0, a_{-1}, 0, a_0, 0, a_1, 0, a_2, 0, ...

Table 5.1 η for duobinary and TPC codes.

DSV	*duobinary*	*TPC*
1	0.500	0.694
2	0.694	0.850
3	0.793	0.910
4	0.850	0.940

This problem is treated more generally. That is, let us consider how to realize the limiting transmission efficiency. The most well-known solution to the problem is thought to be the block coding. When block length n is given, the number of possible block codes, i.e., sequences

which start the initial state and come back to the same state under the restriction of *DSV* after n time units, is calculated by

$$N_B(n) = \mathrm{S}\, A^n\, \mathrm{S}^t \qquad (5.3)$$

where vector S is the initial state row vector such that only one entry corresponding to the initial state is 1 and the other entries are all 0's, and t means the transposition of a matrix. Let us now define the transmission efficiency of block length n, $\eta_B(n)$, by the following equation

$$\eta_B(n) = \frac{\log_2 N_B(n)}{\log_2 N(n)} = \frac{\log_2 N_B(n)}{n} \quad .$$

If a state transition diagram of a composite automaton is aperiodic, then $\lim_{n\to\infty} \eta_B(n)$ is shown to exist and coincides with η, which is previously defined. If it is peri-

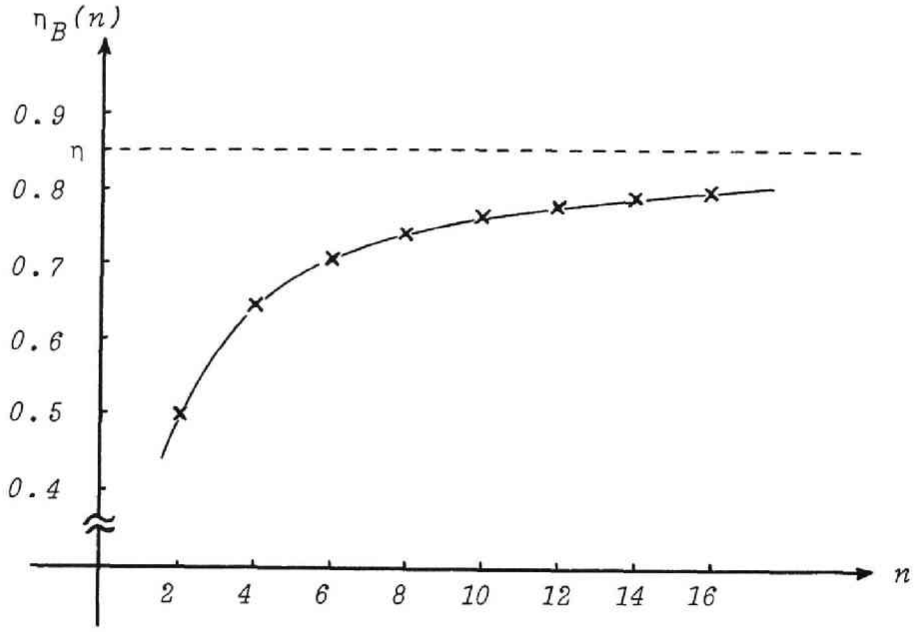


Fig. 5.3 $\eta_B(n)$ curve for TPC code with *DSV*=2.

odic, however, $\lim_{n\to\infty} \eta_B(n)$ does not always exist. If it is assumed that the period is p, then it can be proven that $\lim_{n\to\infty} \eta_B(np)$ always exists and coincides with η.

Again, take Fig.5.2 for the illustration. It is a transition diagram of period 2. So $\lim_{n\to\infty} \eta_B(2n)$ exists and coincides with η. In Fig.5.3, the curve of $\eta_B(2n)$ is plotted against n from 1 to 8. From the curve, it can be observed that it approaches η=0.850 as n tends to infinity.

5.2.2 $(1/2)f_r$-Component Elimination

Transmission codes with a spectral null at $(1/2)f_r$ often require narrower bandwidth than those without it. In addition, they are said to be especially suitable for external timing wave superimposition in conjunction with a high speed PCM communication system. In this section, it is shown that some transmission codes can be made $(1/2)f_r$-component free by properly selecting the set of input sequences, even if they have originally $(1/2)f_r$-component.

Here, we will restate the necessary and sufficient condition for a transmission code to have a spectral null at $f=(1/2)f_r$, which is slightly different from , but equivalent to, that given in Chapter 3. Namely, if we represent a given transmission code by a time series $\{a_i\}$, then the condition is that the newly generated code $\{b_i\}$, defined by $b_i \triangleq (-1)^i a_i$, has a spectral null at dc (f=0). Just like the parameters DS and DSV in the case of dc-elimination, we define weighted DS at time $(2k+1)T$ and weighted DSV at $f=(1/2)f_r$ (abbreviated as $wDS_{(1/2)}$ and $wDSV_{(1/2)}$, respectively) as follows:

$$wDS_{(1/2)} \triangleq \sum_{i=0}^{2k+1} b_i = \sum_{i=0}^{k} (a_{2i} - a_{2i+1})$$

$$wDSV_{(1/2)} \triangleq \max_{k_1 \le k_2} \left| \sum_{i=2k_1}^{2k_2+1} (-1)^i a_i \right| .$$

Namely, $wDS_{(1/2)}$ is defined by the DS during even time intervals of a derived code $\{b_i\}$ and $wDSV_{(1/2)}$ is defined as the difference between the upper and lower bounds of $wDS_{(1/2)}$. The $wDSV_{(1/2)}$ is closely related with the amount of $(1/2)f_r$ spectral component. So it seems reasonable to use $wDSV_{(1/2)}$ as the parameter of $(1/2)f_r$-component elimination. It is thought that $wDSV_{(1/2)}$ is one of the good measures showing how sharply the power spectrum approaches to zero around the neighborhood of $(1/2)f_r$.

The limiting transmission efficiency $\eta_{(1/2)}$ at $f=(1/2)f_r$ under the condition of $(1/2)f_r$-component elimination is defined by

$$\eta_{(1/2)} \triangleq \lim_{n\to\infty} \frac{\log_2 N_R(2n)}{2n} ,$$

where $N_R(n)$ is the number of possible binary input sequences of an encoding automaton satisfying the given $wDSV$ restriction during n-unit time interval.

In order to calculate $N_R(n)$ satisfying given $wDSV$ bound at $f=(1/2)f_r$, we must first construct a finite automaton M', yielding the output sequence $\{b_i\}$, from original encoding automaton M. However, $wDS_{(1/2)}$ and $wDSV_{(1/2)}$ are more easily treated by dealing with the automaton M'' producing the sequence $\{c_i\}$ defined by $c_i \triangleq a_{2i} - a_{2i+1}$. The procedure for constructing M' and M'' from M is roughly illustrated in Fig.5.4(a)∿(d) using an example of a well-known bipolar code shown in Fig.5.4(a).

First, make a copy of M and let the state correspond-

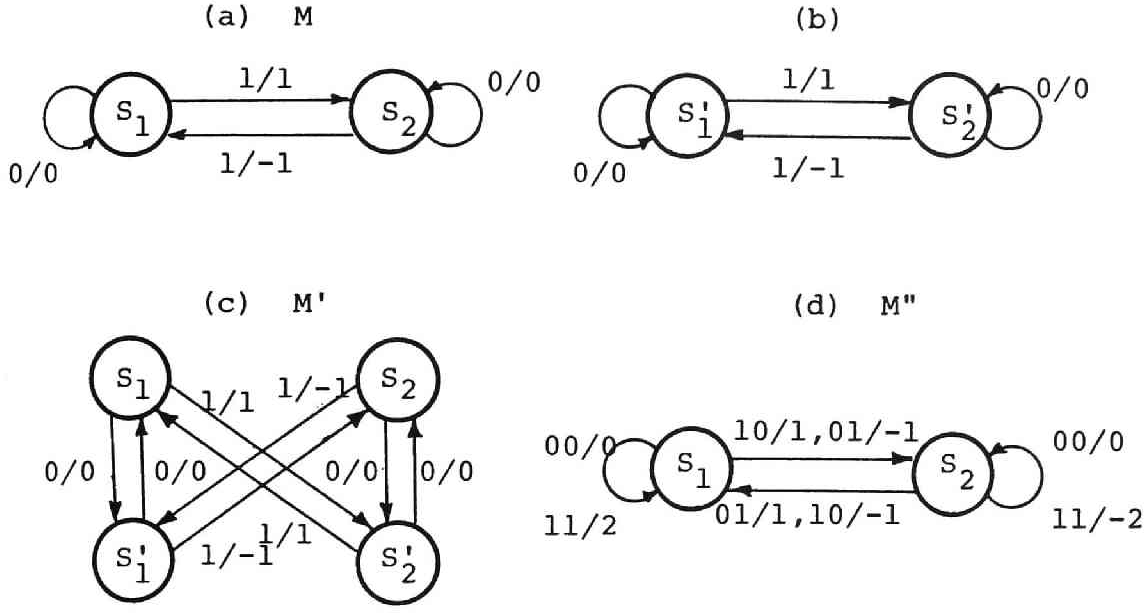


Fig. 5.4
(a) Encoding automaton M of bipolar code and (b) its copy.
(c) Encoding automaton M' producing output sequence $\{(-1)^i a_i\}$, where $\{a_i\}$ is the sequence generated by M.
(d) Finite automaton M" outputting $\{c_i\}=\{a_{2i}-a_{2i+1}\}$.

ing to s_i be s_i' as in Fig.5.4(b). Then replace all the state transitions from s_i to s_j by the corresponding one from s_i to s_j'. Concerning state transitions from s_i' to s_j', they are all replaced by corresponding transitions from s_i' to s_j, and at the same time, associated output is negated with its sign. Resultant automaton M' is shown in Fig.5.4(c). Once M' is obtained, M'' is constructed by letting a set of states be equal to that of M and replacing any state transition in M' by a two step one with associated output being a sum of the corresponding outputs and associated input being a pair of corresponding inputs, which is shown in Fig.5.4(d).

Formally speaking, M'' is constructed from M as $M''=(S,X',Y',\delta',\lambda')$, where

$$X' = X\times X \ , \quad Y'= \{y_1-y_2 \mid y_1\varepsilon Y,\ y_2\varepsilon Y\} \ ,$$
$$\delta'(s,x_1x_2) = \delta(\delta(s,x_1),x_2),$$
and $$\lambda'(s,x_1x_2) = \lambda(s,x_1) - \lambda(\delta(s,x_1),x_2) \ .$$

Once the automaton M'' is constructed, it is not a difficult task to compute $\eta_{(1/2)}$, and is almost the same procedure as in the case of dc-component elimination. Firstly, we must derive a composite automaton M_c satisfying $wDSV_{(1/2)}$ restriction, which is constructed from automaton M'' and the set of $(d+1)$ consecutive integers $D=\{[(1-d)/2],\ldots,-1,0,1,\ldots,[(1+d)/2]\}$ assuming that $wDSV_{(1/2)}=d$. The procedure for constructing M_c is as follows. Namely, M_c is characterized by a quintuple

$$M_c = (S\times D,\ X',\ Y',\ \delta_c,\ \lambda_c) \ ,$$

where

$$\delta_c((s,d),x') = \begin{cases} (\delta'(s,x'),d+\lambda'(s,x')) & \text{if } d+\lambda'(s,x')\varepsilon D \\ \text{undefined} & \text{if } d+\lambda'(s,x')\notin D \ , \end{cases}$$

$$\lambda_c((s,d),x') = \begin{cases} \lambda'(s,x') & \text{if } d+\lambda'(s,x')\varepsilon D \\ \text{undefined} & \text{if } d+\lambda'(s,x')\notin D \ . \end{cases}$$

The derived automaton M_c may not be strongly connected, even if original automaton M'' is. In such a case, most principal strongly connected component with largest information transmission capability should be selected as a representative of M_c. Hereafter M_c is assumed to be chosen as such. To evaluate $N_R(2n)$, it is necessary to define a state transition matrix A_2 of M_c, which corresponds to the incidence matrix of M_c regarded as a directed graph. A state transition matrix A_2 is defined according to the following rule, i.e., its (i,j) entry is equal to the number of input symbols which cause direct

state transition from state i to state j. Then, the number of possible allowed paths of length n in the state transition diagram of M_c, which is equal to those of length $2n$ in the state transition diagram of M, is calculated as

$$N_R(2n) = (0\ldots0\overset{i}{\underset{\smile}{1}}0\ldots0)\ A_2^n \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} ,$$

assuming that the state i is the initial state.

Thus, in order to evaluate $N_R(2n)$, it is necessary to calculate powers of matrix A_2. However, as in the case of dc-component elimination, it is not necessary if the goal is the calculation of $\eta_{(1/2)}$ only. Next useful proposition tells us how to calculate $\eta_{(1/2)}$ efficiently.

[Proposition 5.2] The limiting transmission efficiency $\eta_{(1/2)}$ is equal to $(1/2)\log_2\lambda_{max}^{(2)}$, where $\lambda_{max}^{(2)}$ is the largest positive eigenvalue of the state transition matrix of the composite automaton M_c described above.

Different from the case of dc-component elimination it should be noted that $\eta_{(1/2)}$ is calculated as $\log_2\lambda_{max}^{(2)}$ divided by two, because the matrix A_2 corresponds to the state transition of M observed at every two time units.

To clarify the above discussion, we will give an example based on a bipolar code again. Starting from the M" shown in Fig.5.4(d), M_c is constructed as shown in Fig. 5.5 according to given $wDSV$'s from 1 to 3. As remarked before, there arises two strongly connected components in M_c. However, they are essentially the same and therefore only one of them is illustrated in Fig.5.5.

A state transition matrix A_2 for M_c with $wDSV_{(1/2)}$=2, for instance, is written as

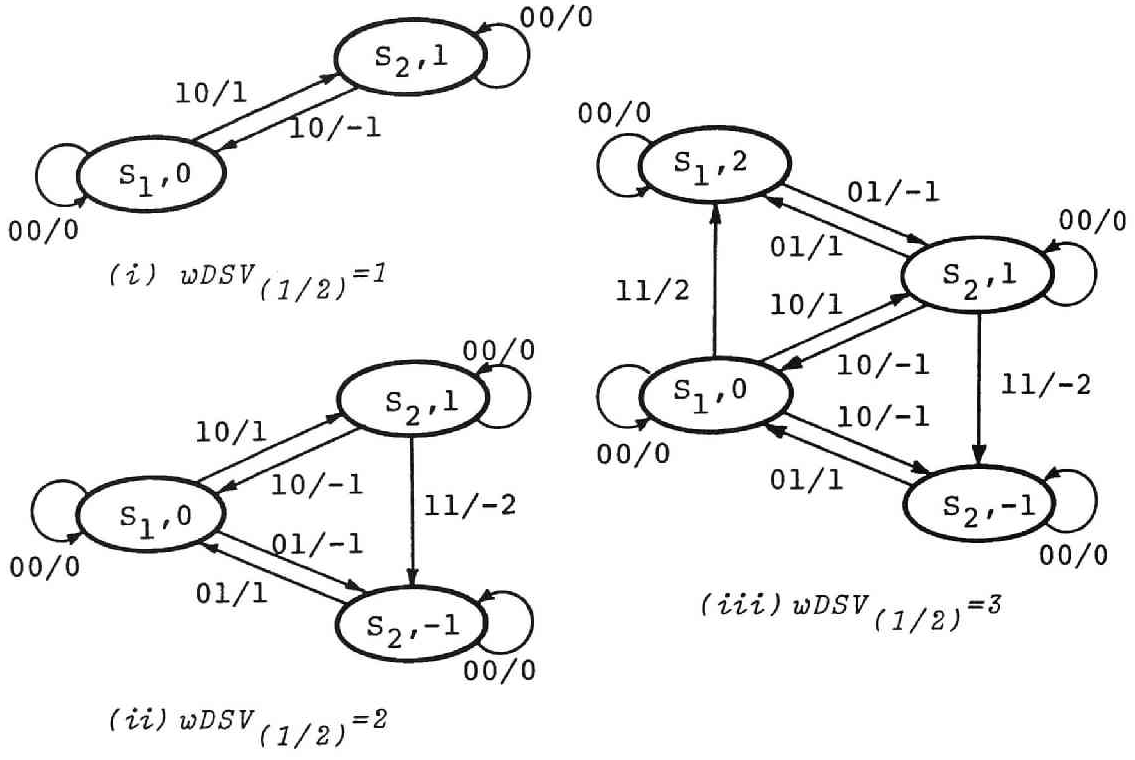


Fig.5.5 Composite automata M_c's for $wDSV$=1∿3.

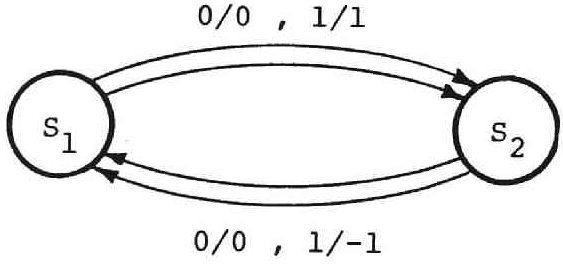


Fig.5.6 Encoding automaton of TPC code.

Table 5.2

$\eta_{(1/2)}$'s for bipolar and TPC codes.

wDSV	bipolar	TPC
1	0.5	0
2	0.694	0
3	0.793	0
4	0.850	0

$$A_2 = \begin{matrix} (s_1, 0) \\ (s_2, 1) \\ (s_2, -1) \end{matrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} .$$

The largest positive eigenvalue $\lambda_{max}^{(2)}$ of A_2 is then computed as $\lambda_{max}^{(2)} = (3+\sqrt{5})/2$. Accordingly $\eta_{(1/2)} \fallingdotseq 0.694$. Computed results of $\eta_{(1/2)}$ for bipolar code with $wDSV$ from 1 to 4 are shown in Table 5.2. For the sake of comparison, $\eta_{(1/2)}$'s for TPC code are also shown to be all zeros. The encoding automaton of TPC code is shown in Fig.5.6. It is shown, in Section 5.2.5, that it is impossible to eliminate $(1/2)f_r$-component by restricting the set of input sequences. We can see from the table that $\eta_{(1/2)}$ for bipolar code is completely the same as previously computed η for duobinary code in the case of dc-component elimination. It can be proven that it holds true for any pair of the same values of $wDSV_{(1/2)}$ for bipolar code and DSV for duobinary code.

5.2.3 $(h/k)f_r$-Component Elimination

Extending the discussion in Section 5.2.1 furthermore, it is shown that $(h/k)f_r$ spectral component can be eliminated for some transmission codes by only restricting the set of input sequences appropriately, where h and k are relatively prime integers. It is already known that one condition for a digital coded sequence $\{a_i\}$ to have a spectral null at $f=(h/k)f_r$ is that a newly derived sequence $\{d_i\}$, defined by $d_i = \sum_{n=0}^{k-1} a_{ki+n} e^{-j2\pi n(h/k)}$,

has a spectral null at dc. So, we shall define $wDS_{(h/k)}$, $wDSV_{(h/k)}$ and $\eta_{(h/k)}$ at $f=(h/k)f_r$ as follows:

$$wDS_{(h/k)} = \sum_{i=0}^{K} d_i \ ,$$

$$wDSV_{(h/k)} = \max_{N<M} \left| \sum_{i=N}^{M} d_i \right| \ ,$$

$$\eta_{(h/k)} = \lim_{n\to\infty} (\log_2 N_R(kn))/kn \ ,$$

where $d_i = \sum_{n=0}^{k-1} a_{ki+n} e^{-j2\pi n(h/k)}$, and $\{a_i\}$ is assumed to be an original transmission code sequence. $N_R(kn)$ is the number of possible input sequences satisfying a given $wDSV_{(h/k)}$ constraint during period of kn.

By deriving an encoding automaton outputting $\{d_i\}$ and then constructing a compositie automaton satisfying a $wDSV_{(h/k)}$ constraint as before, we can easily calculate $\eta_{(h/k)}$ by the following relation,

$$\eta_{(h/k)} = (1/k) \log_2 \lambda_{max}^{(k)} \ ,$$

where $\lambda_{max}^{(k)}$ is the largest positive eigenvalue of a state transition matrix corresponding to the composite automaton.

5.2.4 Calculation of Power Spectrum of Transmission Code Sequences with Restricted Inputs

We have thus far proven that several kinds of power spectral reshapings are possible by restricting the set of input sequences to the encoding automaton. In this section, the power spectral calculation method with speci-

fied input constraint will be given and some calculated results will be illustrated, showing that how the power spectrum is reshaped according to the selected parameters DSV or $wDSV$ varies in the case of spectral elimination at the specific frequency. Simply by power spectra of transmission code sequences with input restriction, we mean the power spectra of the output sequences bearing the maximum amount of information under the given constraint. That is to say, we assume that it is coded by some ideal encoder such that the transmission rate attains the maximum value $\eta_{(1/2)}$ given in Prop.5.2 or $\eta_{(h/k)}$ generally.

Compared with the calculation method without input restriction, the only necessary step which should be changed is the construction of the transition probability matrix P. In Chapter 2, the occurrence probability of an input symbol causing state transition from state i to j, independent of the present state, is assigned to the (i,j) entry, p_{ij}, of matrix P. However, in the case of input restriction, the state transition probability largely depends on the present state and for some states there exists an input symbol not allowed because of the violation of given restriction. The following lemma shows how the state transition probability matrix P is determined.

[Lemma 5.1] [30] Let $A=(a_{ij})$ be the adjacency matrix corresponding to the state transition diagram with input restriction. Then the limiting transmission efficiency (channel capacity) $\eta = \lim_{n\to\infty}(\log_2 N_R(n))/n$ of the encoding automaton with given constraint is attained under the following probability assignment of the state transition probability matrix P, namely its (i,j) entry

$$p_{ij} = a_{ij}x_j/\lambda_{max}x_i$$

where λ_{max} is the largest positive eigenvalue of matrix A

and $\mathbf{x} = (x_1, \ldots, x_n)$ is the corresponding eigenvector whose components are all positive, i.e., $x_i > 0$ $(1 \leq i \leq n)$.

[Proposition 5.3] Power spectra of transmission codes with restricted inputs are calculated by the following method.

Step 1 : Construct the state transition diagram satisfying the given restriction as illustrated in Section 5.2.1.

Step 2 : Let P be the matrix given in Lemma 5.1.

Step 3 : Putting other matrices and vectors in just the same way as described in Chapter 2, calculate power spectrum according to the procedure given in Chapter 2.

Let us illustrate the above procedure concerning *dc*-component eliminated duobinary and TPC codes previously discussed. Let us first take duobinary code with the input restriction $DSV=2$ shown in Fig.5.7. Its state transition matrix A and output matrix Z are readily obtained as follows.

$$A = \begin{array}{c} (s_1, 1) \\ (s_2, 0) \\ (s_1,-1) \\ (s_2, 1) \\ (s_1, 0) \\ (s_2,-1) \end{array} \begin{pmatrix} & & & 1 & 0 & 0 \\ & 0 & & 0 & 1 & 1 \\ & & & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & & & \\ 1 & 1 & 0 & & 0 & \\ 0 & 0 & 1 & & & \end{pmatrix}, \quad Z = \begin{pmatrix} & & & 0 & 0 & 0 \\ & 0 & & 0 & 0 & -1 \\ & & & 0 & 1 & 0 \\ \hline 0 & -1 & 0 & & & \\ 1 & 0 & 0 & & 0 & \\ 0 & 0 & 0 & & & \end{pmatrix}.$$

Then λ_{max} of the matrix A and its corresponding eigenvector $\mathbf{X}$ is calculated as $\lambda_{max}=1.618$ and $\mathbf{X}=(\alpha, 1, 1, 1, 1, \alpha)$, respectively, where $\alpha=0.618$. By Lemma 5.1, the state transition probability matrix P is also determined which achieves the maximum transmission efficiency.

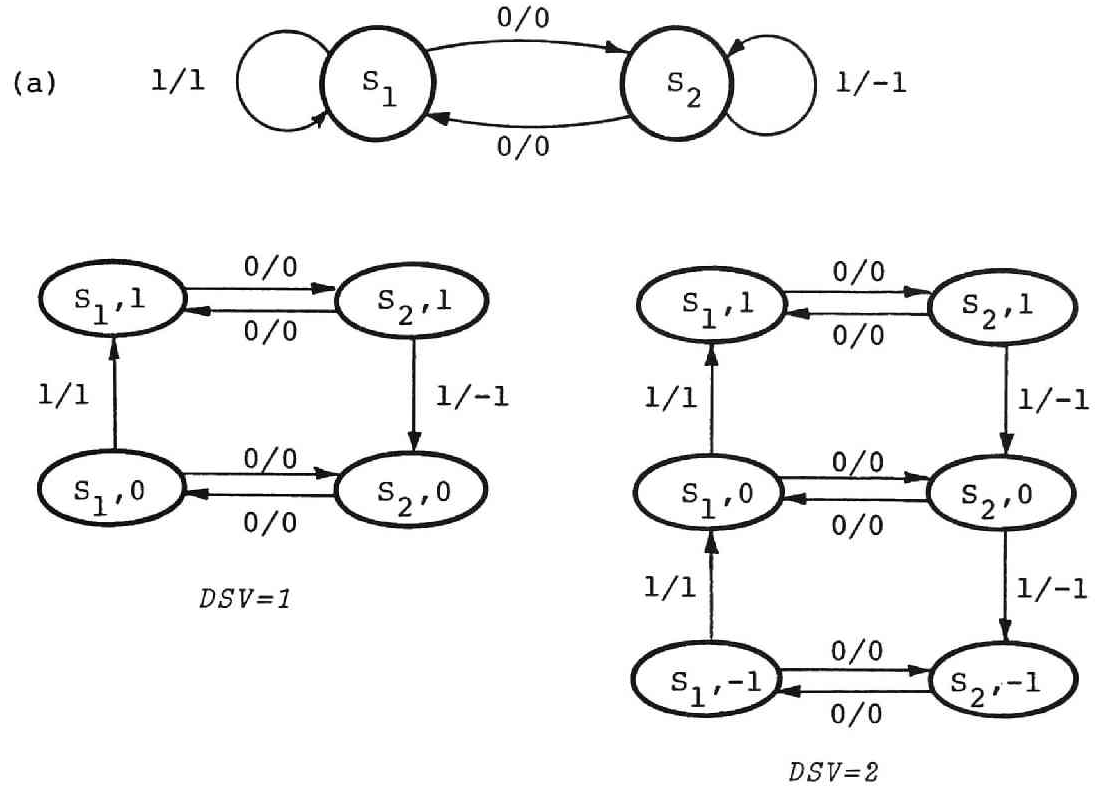


Fig. 5.7 (a) Duobinary code and (b) its composite automata with constraint *DSV*=1 and 2.

$$P = \left(\begin{array}{ccc:ccc} & & & 1 & 0 & 0 \\ & 0 & & 0 & \alpha & \beta \\ & & & 0 & \alpha & \beta \\ \hdashline \beta & \alpha & 0 & & & \\ \beta & \alpha & 0 & & 0 & \\ 0 & 0 & 1 & & & \end{array}\right),$$

where α=0.618 and β=0.382. We can then compute power spectrum of duobinary code with input constraint *DSV* =2. Calculated normalized power spectra of duobinary code with *DSV*=1,2,3 and 4, together with that of without restriction, are plotted in Fig.5.8. It is clear from the figure that *dc*-component is completely suppressed as expected when *DSV*

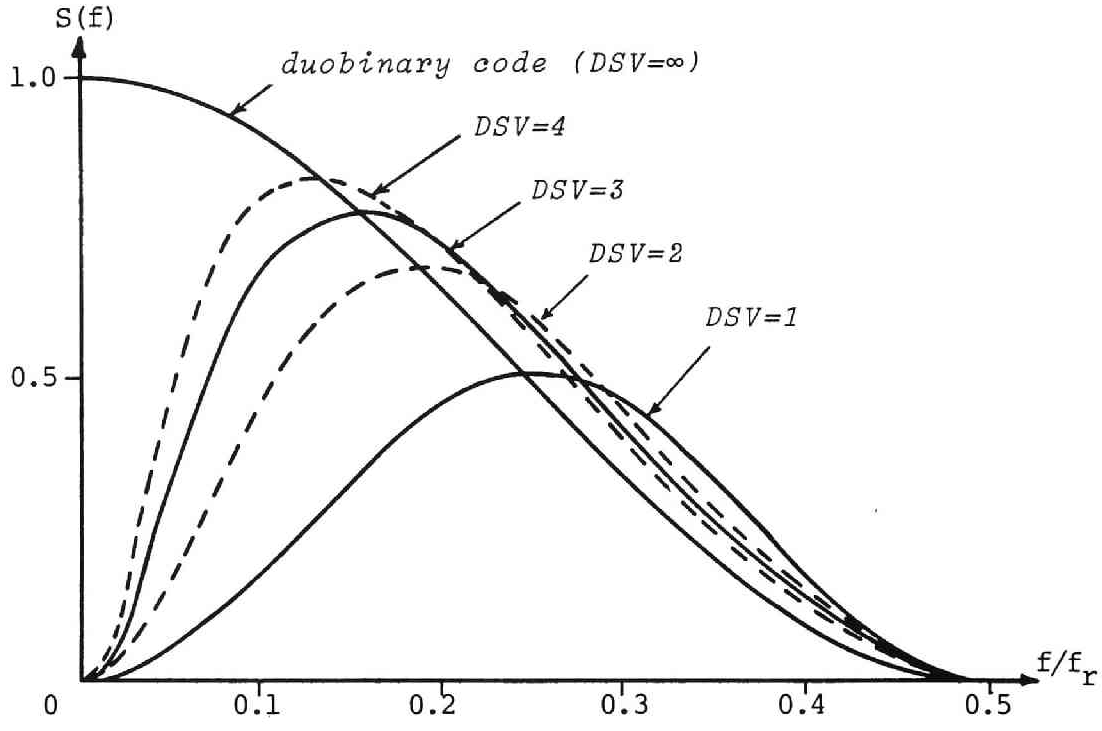


Fig. 5.8 Normalized power spectra of duobinary code with $DSV=1\sim4$.

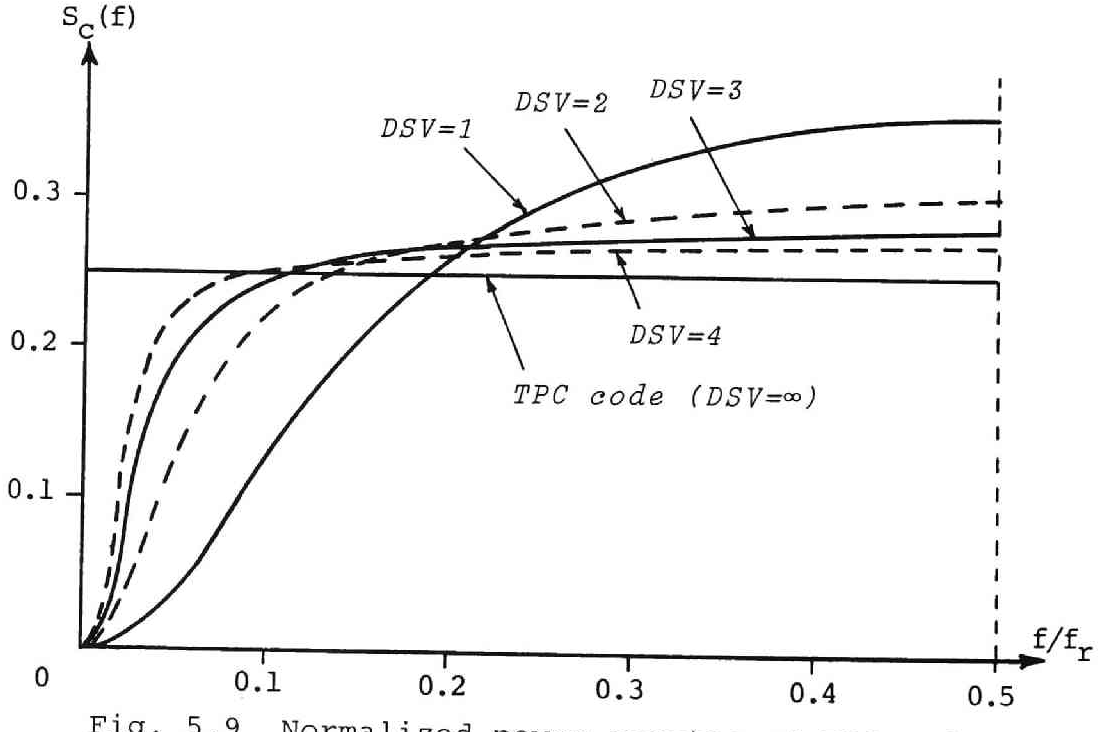


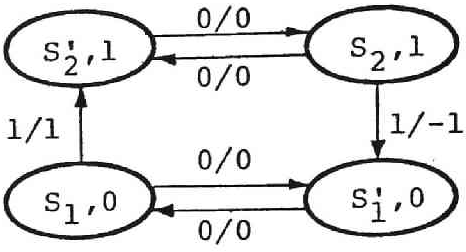
Fig. 5.9 Normalized power spectra of TPC code with $DSV=1\sim4$.

constraint is given, as opposed to the original duobinary code without DSV constraint. In addition, the slope of the curve in the neighborhood of spectral null becomes sharper and sharper as the DSV increases, which means the increase of low frequency component.

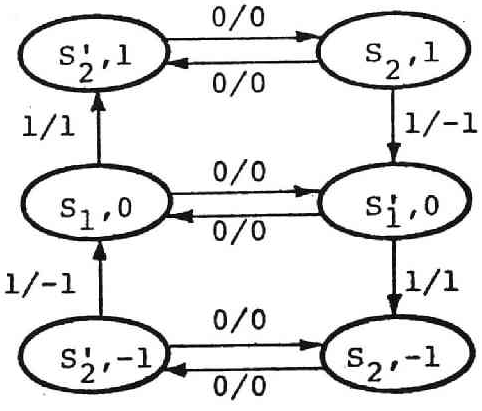
The circumstances are very similar in the case of TPC code and the computed curves for DSV ranging from 1 to 4 are plotted in Fig.5.9.

Next, we shall calculate power spectrum of $wDSV_{(1/2)}$ restricted bipolar code treated in the preceding section. It is calculated by first constructing state transition diagrams with DSV, equivalent to given $wDSV_{(1/2)}$, restriction based on that of the sequence $\{b_i\}$ derived from original sequence $\{a_i\}$ by putting $b_i=(-1)^i a_i$. Once it is constructed, then modify it to obtain the state transition diagram of $wDSV_{(1/2)}$ restricted original bipolar code by changing that sign of outputs which is altered when constructing the sequence $\{b_i\}$ from $\{a_i\}$. That is, modification in this case means simply changing the sign of the outputs associated with the outgoing edge (state transition) from (s_i', d) for all i and d.

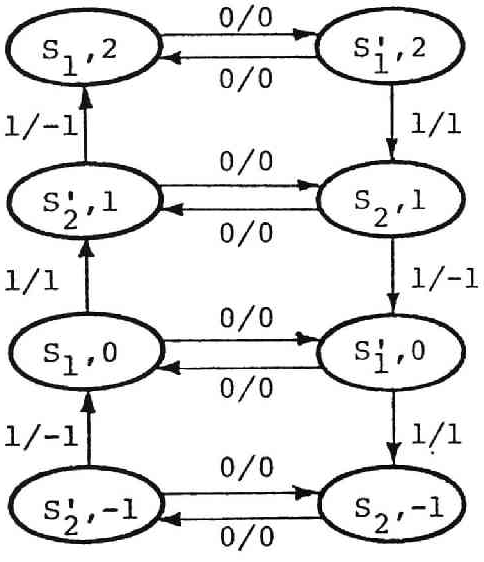
Based on the above procedure, we have constructed the state transition diagram of $wDSV_{(1/2)}$ constraint bipolar code. They are shown for $wDSV_{(1/2)}$ ranging from 1 to 3 in Fig.5.10. Although some of the automata thus obtained should be state minimized, we leave them undone for the convenience of comparison between them. Once the state transition diagrams of $wDSV_{(1/2)}$ constraint bipolar code are obtained, the power spectra of sequences generated by them are readily calculated by Prop.5.3. They are drawn in Fig.5.11. They are really symmetric with the curves in Fig.5.8 with respect to the axis $f=(1/2)f_r$.



wDSV=1



wDSV=2



wDSV=3

Fig.5.10 Composite automata with $wDSV_{(1/2)}$=1~3 derived from bipolar code.

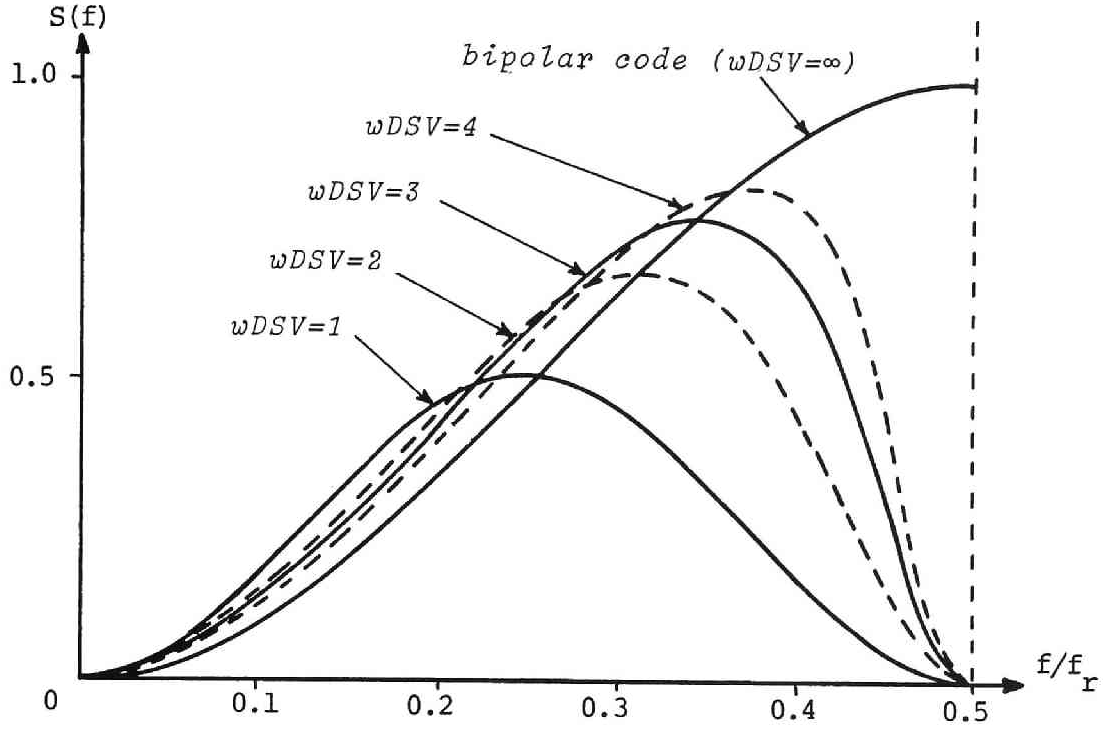


Fig. 5.11 Normalized power spectra of bipolar code with $wDSV_{(1/2)}=1\sim4$.

5.2.5 A Condition of Spectral Reshaping

We have thus far described that some transmission codes can be restricted with their inputs such that resultant power spectra have nulls at submultiples of f_r. However, it is also shown that there exist transmission codes for which it is impossible to give such spectral reshapings. In this section, we shall examine in what circumstances it is possible for a given transmission code sequence to have such spectral properties. Since the general discussion about $(h/k)f_r$-component elimination is

reduced to that of *dc*-component elimination if we derive a state transition diagram of the sequence $\{d_i\}$ defined by

$$d_i = \sum_{n=0}^{k-1} a_{ki+n} e^{-j2\pi n(h/k)}$$

where $\{a_i\}$ is the original transmission code sequence, we shall focus our discussion on that of *dc*-component elimination.

[Proposition 5.4] If a given transmission code never has a line spectrum at *dc* (f=0), then it is sure that η is positive when a sufficiently large *DSV* is given. Namely, *dc*-component elimination is surely realized at some sacrifice of transmission rate.

(Proof) Omitted.

Next, a necessary and sufficient condition for a given transmission code to have a spectral null at *dc* is stated in terms of closed paths in the state transition diagram of an encoding automaton.

[Proposition 5.5] When a transmission code is given, it can be made *dc*-component free, i.e., $\eta > 0$ when a sufficiently large *DSV* is given, if and only if there exist a pair of different closed paths with a common state, whose outputs are summed up to be zero, in its state transition diagram of the encoding automaton. By different paths, we mean they are distinguished by their input-output relations.

(Proof) Omitted.

[Corollary 5.1] If there exist two closed paths, not necessarily with a common state, whose outputs along them are summed up to be positive for one and negative for the

other in the strongly connected state transition diagram of an encoding automaton, then dc-component elimination is possible , i.e., η is positive.

As an illustration, a TPC code is shown to be impossible to have a spectral null at $f=(1/2)f_r$ by deriving a state transition diagram producing $\{d_i\}$ $(d_i=a_{2i}-a_{2i+1})$ shown in Fig.5.12. That is, there exist no two different closed paths satisfying the condition in it. Therefore, we can conclude that $\eta_{(1/2)}$ for TPC code equals to zero and $(1/2)f_r$-component elimination is impossible.

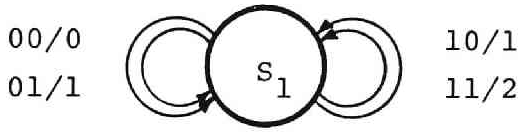


Fig. 5.12 Encoding automaton producing sequence $\{d_i\}$ $(d_i=a_{2i}-a_{2i+1})$ derived from TPC code sequence $\{a_i\}$.

5.3 Generation of Discrete Line Spectra

Even if a transmission code does not have line spectra originally, it may generate them if applied input sequences are properly restricted. For instance, an even or odd parity check code which we frequently encounter in various places is shown to be a typical example yielding line spectra when it is bipolar coded. Such line spectra are useful as a timing information source and can also be used as a measure of how it is likely to have line spectra by input probability deviation from the statistical independent assumption.

An encoding automaton with an aperiodic transition diagram is known to have line spectrum at f=0 if and only if an ensemble average m_i of the output sequences is not equal to zero. In general, Eq.(2.14), discussed in Chapter 2, shows under what circumstances line spectra will appear when an arbitrary encoding automaton is given with periodic state transition diagram of period N. It is concluded that for a given code to yield line spectra at frequencies $f=(1/2)f_r$ or $(1/3)f_r$ it is necessary for a corresponding state transition diagram to have a period of multiples of 2 or 3, respectively. In other words, to yield line spectra by restricting inputs, it is necessary to introduce the periodicity into the state transition diagram of an encoder. As an illustrative example for introducing the periodicity, we will restrict the set of input sequences of bipolar encoding automaton such that a symbol zero must be followed by another redundant zero symbol. That is to say, source input sequence is encoded such that symbol 1 and 0 correspond to 1 and 00, respec-

tively. Such input restriction transforms the original bipolar encoding automaton as shown in Fig. 5.13 by augmenting two states s_1' and s_2'. It is assumed that s_1 or s_2 is the initial state. The period of the state transition diagram is two. It is easily shown that the set of states S can be divided as $Q_1 \cup Q_2$, where $Q_1=\{s_1, s_1'\}$ and $Q_2=\{s_2, s_2'\}$. Moreover, it is easy to prove that m_1 is positive, whereas m_2 is negative. This means that the automaton produces line spectrum at $f=(1/2)f_r$ from Eq. (2.14). Intuitively, starting from an initial state,

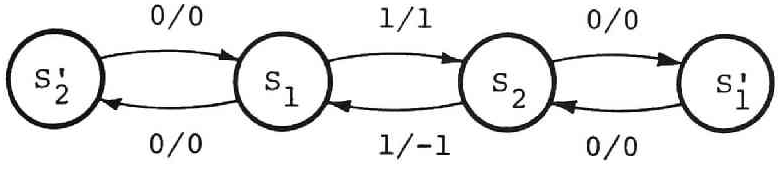


Fig. 5.13 Encoding automaton derived from bipolar code with odd number of consecutive zeros inhibited.

say s_1, at time zero, two consecutive zero input will cause the state transition such that positive output, +1, is surely generated at even time slot, whereas negative one, -1, is always at odd time slot. It is this regularity that will cause to appear line spectra. Such restriction lowers the information transmission efficiency to about 0.694.

If we apply the same input restriction to the duobinary code, it is shown that line spectrum at $f=0$ will appear. This is verified by the fact that the corresponding output for input one has always the same polarity.

Since the restriction is imposed on the input sequence only, it is true that if a given code does not have any spectral component at $f=(h/k)f_r$, then it will never have line spectra even if whatever restriction is applied to

the input. Thus, generation of line spectrum requires at least existence of continuous component at the same frequency.

[Proposition 5.7] Let a transmission code with restricted input be given. The condition for it to have line spectra at $f = (h/k) f_p$ is that the corresponding state transition diagram satisfying the restriction have a period of multiple of k, say kl, and at the same time the equation $\sum_{I=0}^{kl-1} m_I e^{-j2\pi(h/k)I} \neq 0$ holds true.

As a more systematic procedure for introducing a periodicity into a state transition diagram, a block coding is thought to be the simplest of all. We deal with a parity check code of block length n. The last digit of a block is assumed to be a parity bit which is equal to mod 2 sum of all the remaining n-1 digits (even parity) or its complement (odd parity). If we input the parity check code sequence into a bipolar encoding automaton, it behaves as if it has a period of n or $2n$ according to the even or odd parity, respectively. This is due to the fact that bipolar encoding automaton (Mealy-type) changes its state whenever input symbol "1" is applied and the number of input symbol "1" in a single block is fixed to be even or odd in a parity check code. Therefore, in case of even parity check code, for instance, parity check bit at the tail end of a block forces the state transition to the initial state where it was at the biginning of the block. Thus at the biginning of each block, it is always in the same initial state in case of even parity, while it changes the state regularly, namely it behaves like with period $2n$, in case of odd parity. The state transition is completely characterized by a

state trasition diagram with the number of states $2n$-1 or twice of it. For example, state transition diagrams of even and odd (5,4) parity check code are shown in Fig. 5.14(a) and (b), respectively. Based on them, we will calculate power spectrum of bipolar coded (5,4) even parity check code. The set of states S is divided as $S=Q_1 \cup Q_2 \cup Q_3 \cup Q_4 \cup Q_5$, where $Q_1=\{s_1\}$, $Q_2=\{s_2,s_3\}$, $Q_3=\{s_4,s_5\}$, $Q_4=\{s_6,s_7\}$, and $Q_5=\{s_8,s_9\}$. Assuming that occurrence probabilities of input symbols 1 and 0 are both equal to 0.5, we will calculate local average m_I which is defined to be an average over subset Q_I. From Fig.5.14(a), it is apparent that $m_1=0.5$, $m_2=m_3=m_4=0$, and $m_5=-0.5$. Accordingly

$$w_d(f)=(1/50)f_r^2|G(f)|^2\sum_{k=-\infty}^{\infty}(1-\cos\frac{2k}{5}\pi)\delta_d(f-\frac{k}{5}f_r).$$

In Fig.5.15, the line spectrum is shown by vertical arrows whose height is proportional to the line spectral quantity together with the continuous spectrum. The continuous power spectrum is easily calculated using the matrix based method. It is shown that the continuous component of $(n,n-1)$ parity check code becomes $(n-1)/n$ times that of normal input sequence without input restriction. In case of (5,4) odd parity check code, line spectrum occurs at the different frequencies from even parity check code, namely

$$w_d(f)=(1/50)f_r^2|G(f)|^2\sum_{k=-\infty}^{\infty}(1-\cos\frac{2k+1}{5}\pi)\delta_d(f-\frac{2k+1}{5}f_r),$$

while the continuous spectrum remains the same as Fig.5.15.

The power spectra of parity check codes are further discussed from more general point of view in the next chapter.

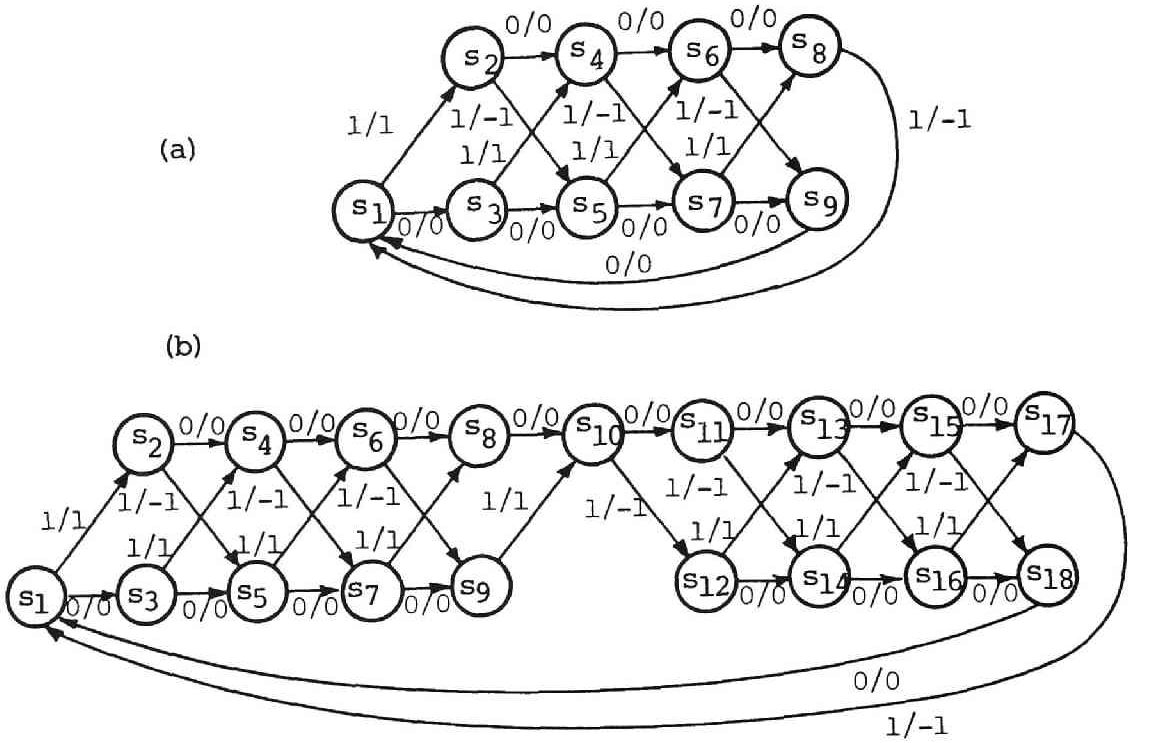


Fig.5.14 State transition diagrams of encoding automata of bipolar coded (a) even and (b) odd (5,4) parity check codes.

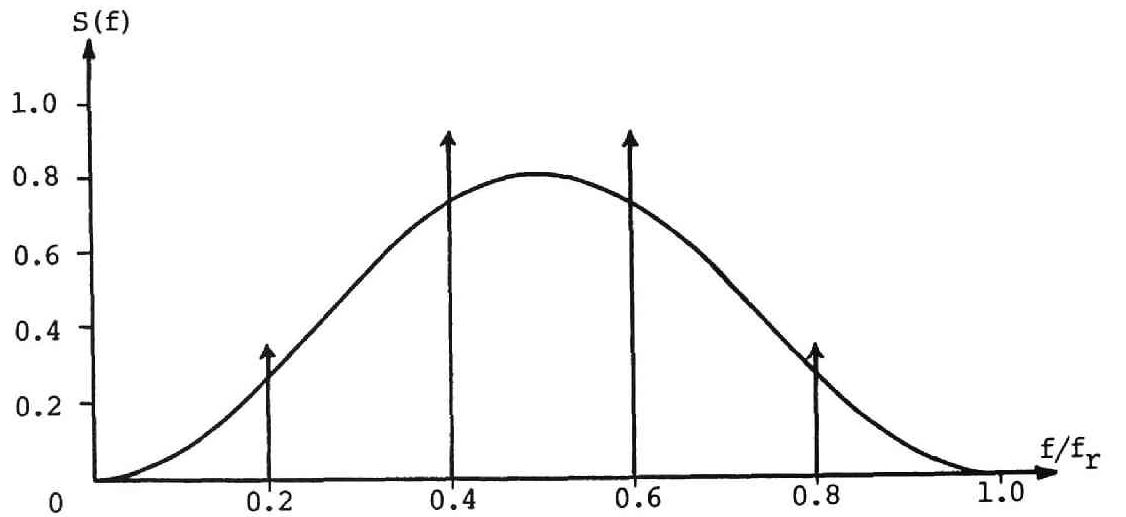


Fig. 5.15 Normalized power spectrum of bipolar coded (5,4) even parity check code.

CHAPTER 6 POWER SPECTRA OF ERROR CONTROL CODE SEQUENCES

As a special case of restricted inputs, error control coded sequences are treated in this chapter. A simple power spectral computation algorithm applicable not only to a unipolar coded error control code but also to a bipolar coded one is presented.

6.1 INTRODUCTION

Recently, error control codes with error detecting or correcting capability tend to be used more and more frequently. Therefore, it is important to develop a power spectral calculation method for a digital signal with its input being an error control coded sequence.

With respect to the power spectrum calculation of error control code, almost nothing seems to have been done, thus far, except for a work by J.H.Gilchrist et al. [21]. They have shown how to calculate a power spectrum of a PAM (pulse amplitude modulation) coded error control code, where the amplitude is directly proportional to the element of GF(q). Their method is based on the direct calculation of the autocovariances and is rather complex because many matrices as well as four dimensional array are needed in the calculation process.

In this chapter, we will present a very simple algo-

rithm of power spectral calculation for an error control coded sequence, which is applicable to a bipolar coded error control code as well as a unipolar coded one, in the case of binary input symbol. Namely, restricting the finite field, over which error control codes are constructed, to GF(2), the amount of computation is significantly reduced. It is also directly translated into a computer program. Major part of the calculation is that of joint probabilities between two or three stochastic variables. It is computed simply by performing row operations against the parity check matrix corresponding to a given error control code in the case of linear codes.

6.2 Error Control Codes

A linear error control code is thoroughly characterized by its generator polynomial or equivalent parity check matrix, even if it is a multiple error correcting code or simply an error detecting parity check code. Here we prefer the latter to the former. A block error control code called (N,K) code will be dealt with in this chapter. That is, a binary information source sequence is firstly separated into blocks of consecutive K digits, and then each block is supplemented with $(N-K)$ check digits to form a complete single block of N digits. We denote the r-th block consisting of binary information digits as a column vector $\mathbf{I}^{(r)}$:

$$\mathbf{I}^{(r)} = (c_1^{(r)}, c_2^{(r)}, \ldots , c_K^{(r)})^t, \qquad (t: \text{transposition})$$

and a complete codeword, supplemented with check digits $c_{K+1}^{(r)}, \ldots, c_N^{(r)}$, as

$$\mathbb{C}^{(r)} = (\underbrace{c_1^{(r)}, c_2^{(r)}, \ldots, c_K^{(r)}}_{\text{information digits}}, \underbrace{c_{K+1}^{(r)}, \ldots, c_N^{(r)}}_{\text{check digits}})^t ,$$

where $c_i^{(r)} \in \{0,1\}$ $(1 \le i \le N)$ for any r. If it is unnecessary to specify the block number, we shall omit the superscripts and adopt the notation like $\mathbb{I}$ and $\mathbb{C}$ for information block and codeword, respectively.

As an example, a well-known (7,4) single error correcting Hamming code is represented as

$$\left(\begin{array}{cccc:ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ \hdashline c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} ,$$

where $c_1 \sim c_4$ and $c_5 \sim c_7$ are information and check digits, respectively. However, we are to adopt another way of representation for the convenience of describting an algorithm stated in a later section. Namely,

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ \hdashline c_5 \\ c_6 \\ c_7 \end{pmatrix} = \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hdashline 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array}\right) \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} .$$

Generally speaking, when an error control code is given in the form of $\mathbb{H}\mathbb{C}=0$, an $\mathbb{H}$ matrix is not necessarily in the form as

$$\mathbb{H} = [\, H_{N-K,K} \,\vdots\, I_{N-K,N-K} \,] , \qquad (6.1)$$

where $H_{N-K,k}$ is an $(N-K)\times K$ matrix and $I_{N-K,\ N-K}$ is an $(N-K)\times(N-K)$ square identity matrix. However, by performing elementary row operations from the left, it can be equivalently transformed into the expression (6.1).

Then, we define a parity check matrix $\mathbb{H}'$ by the following $N\times K$ matrix as

$$\mathbb{H}' = \begin{pmatrix} \mathbf{I}_{K,K} \\ \cdots\cdots \\ H_{N-K,K} \end{pmatrix} \qquad (6.2)$$

where $H_{N-K,K}$ is the same matrix as that in (6.1). As is clear from the above discussion, the next relation holds true:

$$\mathbb{C} = \mathbb{H}' \cdot \mathbf{I}$$

As another example, we illustrate the parity check matrix of a (15,7) double error correcting code which appeared on page 125 in ref.[31], whose generator polynomial g(x) is $(x^4+x+1)(x^4+x^3+x^2+x+1)$, as a preparation for the later calculation of power spectrum. In ref. [31], $\mathbb{H}$ matrix is not given in the form of (6.1), so we need appropriate row operations from the left. The resultant $\mathbb{H}$ matrix, whose column vectors are rearranged in reverse order to meet the representation adopted here, is

$$\left(\begin{array}{ccccccc:cccccccc}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right)$$

$$\longleftarrow H_{8,7} \longrightarrow \quad \longleftarrow I_{8,8} \longrightarrow$$

We call the $\mathbb{H}'$ matrix of the form (6.2), derived from the

above matrix, the parity check matrix of a (15,7) double error correcting code.

Generally speaking, even if information source is stationary, the sequence supplemented with check digits is not stationary, but the ensemble statistics vary periodically with time. That is, an error control code can be viewed as a cyclostationary process [22] with period equal to its block length. Thus, in order to calculate the power spectrum of an error control code, we must use formulas (2.14) and (2.15), discussed in Chapter 2.

Different from expression (2.1), a block coded pulse sequence is more conveniently expressible, in some cases, by

$$x(t) = \sum_{r=-\infty}^{\infty} \sum_{i=1}^{N} a_i^{(r)} g(t-(rN+i)T) \quad . \tag{6.3}$$

where $a_i^{(r)}$ is the output value of transmission code corresponding to $c_i^{(r)}$ and N is assumed to be the block length. Clearly, the relation

$$a_i^{(r)} = a_{rN+i} \tag{6.4}$$

holds. Hereafter, we will make a suitable choice between expressions (2.1) and (6.3) according to circumstances.

For the convenience in later sections, we also define

$$r_i(k) = \text{av}\ (a_i a_{i+k}) \qquad (i\text{: fixed}),$$

$$\overline{r}_i(k) = \text{av}\ (a_i a_{i+k}) - m_i m_{i+k\,(\text{mod}N)} \quad .$$

Then, we have

$$\overline{R}(k) = R(k) - \frac{1}{N} \sum_{i=1}^{N} m_i m_{i+k} = \frac{1}{N} \sum_{i=1}^{N} \overline{r}_i(k) \quad . \tag{6.5}$$

6.3 Power Spectra of Unipolar Coded Error Control Code Sequences

We shall first treat a unipolar coded error control code sequence. It is the simplest code of $a_i^{(r)}$ being equal to $c_i^{(r)}$. We assume that information source digits are stationary and mutually independent. By putting

$$p_i \triangleq \Pr(c_i^{(r)} = 1), \qquad (1 \le i \le N)$$

which is independent of r, local average m_i is readily obtained, i.e.,

$$m_i = \mathrm{av}\{a_i^{(r)}\} = p_i \qquad (1 \le i \le N) \tag{6.6}$$

This implies it is only necessary to know p_i $(1 \le i \le N)$ to calculate a discrete line spectrum of a unipolar coded error control code, from Eq. (2.14). However, p_i's are easily calculated, because c_i can be always represented as a mod 2 sum of independent information digits $c_1, \ldots, c_K$.

[LEMMA 6.1] [32-33] Let c_{i1}, $c_{i2}, \ldots, c_{in}$ be mutually different n independent variables such that $\Pr(c_{ij}=1)=p$ for $1 \le j \le n$. Then

$$f(n) \triangleq \Pr(c_{i1} \oplus c_{i2} \oplus \ldots \oplus c_{in} = 1) = \{1-(1-2p)^n\}/2. \tag{6.7}$$

Lemma 6.1 shows very interesting fact that $\Pr(c_{i1} \oplus c_{i2} \oplus \ldots \oplus c_{in}=1)$ is completely determined by the number of variables n, and not depend how we choose n independent variables.

[Proposition 6.1] If we let a Hamming weight of the

i-th row vector of parity check matrix H' be h_i, then

$$p_i = f(h_i) = \{1-(1-2p)^{h_i}\}/2 \quad .$$

Next, a continuous component of power spectrum is considered. It is important to note that in a unipolar code every two digits belonging to different blocks are independent. This makes the calculation of autocovariance be a very easy task. Namely,

$$\bar{R}(k) = \begin{cases} (1/N)\sum_{i=1}^{N-k}\{Pr(c_i=1 \wedge c_{i+k}=1) - p_i p_{i+k}\} & (0 \le k \le N-1) \\ 0 & (N \le k) \quad . \end{cases} \tag{6.8}$$

The calculation of a joint probability $Pr(c_i=1 \wedge c_j=1)$ is carried out easily by regarding the condition within the parenthesis as a logic function of independent logical variables, i.e., information digits in this case. That is, we first group the information digits into three classes A, B and D, shown in Fig.6.1(a), which correspond to sets of variables appearing in only c_i or c_j or both c_i and c_j, respectively. Then by denoting the mod 2 sum of variables in A, B and D as x_1, x_2 and x_3 (see Fig.6.1(b)), respectively, we can derive the next minterm Boolean expression.

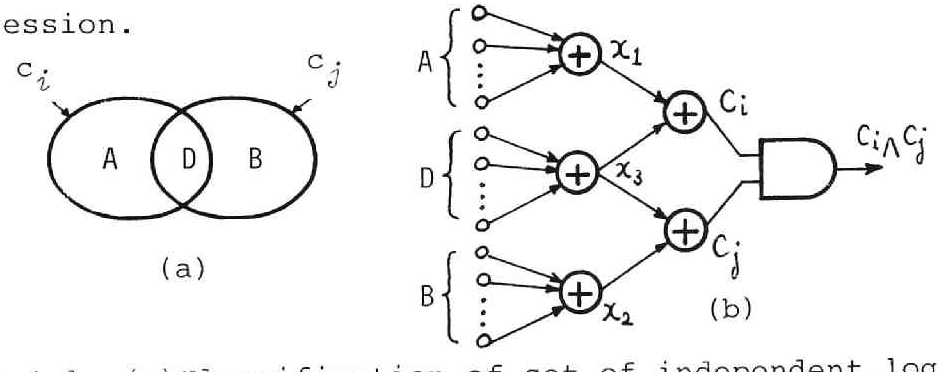


Fig.6.1 (a) Classification of set of independent logical variables (information digits) appearing in c_i and c_j,
(b) logic function representation of $c_i \wedge c_j$.

$$c_i \wedge c_j = (x_1 \oplus x_3) \wedge (x_2 \oplus x_3) = x_1 \wedge x_2 \wedge \bar{x}_3 + \bar{x}_1 \wedge \bar{x}_2 \wedge x_3 .$$

If we let the number of elements in A, B and D be a, b, d, respectively, we have $\Pr(x_1=1)=f(a)$, $\Pr(x_2=1)=f(b)$ and $\Pr(x_3=1)=f(d)$ from Lemma 6.1. Then, by substituting $\Pr(x_i=1)$ to x_i and $1-\Pr(x_i=1)$ to $\bar{x}_i$ in each minterm, we can readily obtain the next result.

$$\begin{aligned}\Pr(c_i \wedge c_j=1) &= f(a)f(b)(1-f(d))+(1-f(a))(1-f(b))f(d) \\ &= f(a)f(b)+(1-f(a)-f(b))f(d) .\end{aligned}$$

The calculation process of a joint probability is summarized in terms of $\mathbb{H}'$ matrix as follows.

[Proposition 6.2] Under the condition that c_i and c_j belong to the same block, $\Pr(c_i \wedge c_j=1)$ is calculated by the following steps.

Step 1: Let the i-th and j-th row vectors of $\mathbb{H}'$ matrix, corresponding to c_i and c_j, be $\mathbb{h}_i$ and $\mathbb{h}_j$ respectively, and bitwise product of them be $\mathbb{h}_i \wedge \mathbb{h}_j$.

Step 2: Let the Hamming weights of $\mathbb{h}_i$, $\mathbb{h}_j$ and $\mathbb{h}_i \wedge \mathbb{h}_j$ be h_i, h_j and $h_{i\cdot j}$ respectively.

Step 3:
$$\begin{aligned}\Pr(c_i \wedge c_j=1) = {} & f(h_i-h_{i\cdot j})f(h_j-h_{i\cdot j})+ \\ & (1-f(h_i-h_{i\cdot j})-f(h_j-h_{i\cdot j}))f(h_{i\cdot j}) .\end{aligned} \tag{6.9}$$

Now we introduce an autocovariance matrix $B=(b_{ij})$ where $b_{ij}=\overline{r}_i(j-i)=\Pr(c_i^{(r)}=1 \wedge c_j^{(r)}=1)-p_i p_j$ for $i \le j$ and $b_{ij}=0$ for $i > j$. Then $\overline{R}(k)$ is obtained by

$$\overline{R}(k) = \begin{cases} (1/N)\sum_{i=0}^{N-k} b_{i,i+k} & (0 \le k \le N-1) \\ 0 & (N \le k) \end{cases} \tag{6.10}$$

Based upon the above procedure, let us calculate a power spectrum of (7,4) Hamming code whose parity check matrix is

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \\ \hline \mathbf{h}_5 \\ \mathbf{h}_6 \\ \mathbf{h}_7 \end{matrix}$$

As an illustration to calculate the matrix B, let us evaluate $b_{5\cdot6}$. As $\mathbf{h}_5=(1110)$, $\mathbf{h}_6=(0111)$ and $\mathbf{h}_{5\cdot6}=(0110)$, we obtain $h_5=3$, $h_6=3$, and $h_{5\cdot6}=2$. Consequently,

$$\begin{aligned} b_{5\cdot6} &= \Pr(c_5 \wedge c_6=1) - p_5 p_6 \\ &= f(1)^2+(1-2f(1))f(2)-f(3)^2 \\ &= 2p(1-p)(1-2p)^2(2p^2-2p+1) \ . \end{aligned}$$

Thus we obtain

$$B = \begin{pmatrix} p-p^2 & 0 & 0 & 0 & \alpha & 0 & \alpha \\ & p-p^2 & 0 & 0 & \alpha & \alpha & \alpha \\ & & p-p^2 & 0 & \alpha & \alpha & 0 \\ & & & p-p^2 & 0 & \alpha & \alpha \\ & & 0 & & \gamma & \beta & \beta \\ & & & & & \gamma & \beta \\ & & & & & & \gamma \end{pmatrix}$$

where $\alpha=p(1-p)(1-2p)^2$

$\beta=2p(1-p)(1-2p)^2(2p^2-2p+1)$,

$\gamma=f(3)-f(3)^2=\{3p(1-p)^2+p^3\}\{(1-p)^3+3p^2(1-p)\}$.

Accordingly, from Eq.(6.10), we have

$$\bar{R}(0)=(1/7)\{4p(1-p)+3f(3)(1-f(3))\}, \quad \bar{R}(1)=(2/7)\beta,$$
$$\bar{R}(2)=(1/7)(2\alpha+\beta), \quad \bar{R}(3)=(3/7)\alpha, \quad \bar{R}(4)=(2/7)\alpha \ ,$$
$$\bar{R}(5)=\bar{R}(6)=(1/7)\alpha, \quad \bar{R}(7)=\bar{R}(8)=\ldots=0.$$

We will show the curves of normalized power spectrum $S_c(f)$ plotted against f/f_r for p= 0.1, 0.3, 0.5, 0.7 and 0.9 in Fig.6.2, where

$$S_c(f) = w_c(f)/f_r |G(f)|^2 = \overline{R}(0)+2\sum_{k=1}^{6}\overline{R}(k)\cos k\omega T \ .$$

Apparently, we notice from the figure that the curve for p=0.5 completely coincides with that of usual unipolar code with normal independent input. This is true for any linear error control code. Except for the curve of p=0.5, *dc* component is increased compared with usual unipolar code.

Another important fact with regard to power spectra of block error control code is that permutation of digits within the block will lead to different spectral shaping, as is referred in [21]. However, spectral properties which are independent of statistics of input sequences, such as having spectral nulls, are of course invariant. In addition, it is shown that some properties such as presence or absence of line spectra or whole power spectrum in the case of p=0.5 are invariant under any permutations within the block. Detailed discussion is left for the future study.

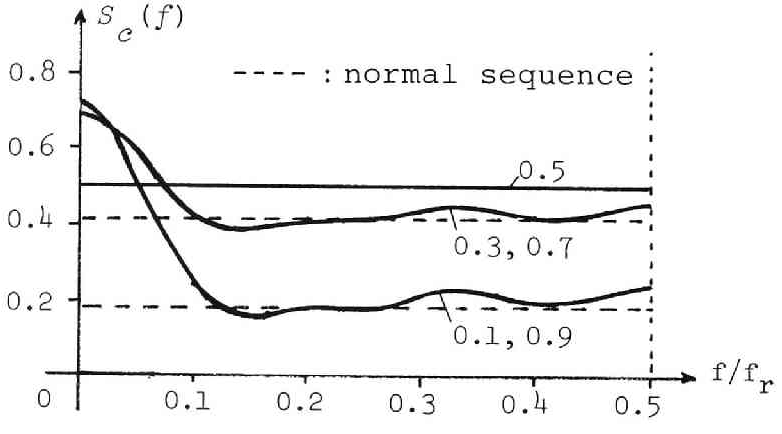


Fig.6.2 Normalized power spectrum $S_c(f)$ of unipolar coded (7,4) Hamming code.

6.4 Power Spectra of Bipolar Coded Error Control Code Sequences

A bipolar code is one of the most widely used transmission code. So it is interesting to calculate the power spectrum of a bipolar coded error control code sequence. An encoding law is accurately described by a Mealy-type automaton shown in Fig.6.3.

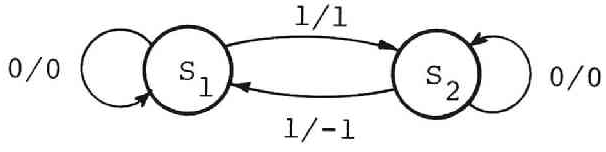


Fig.6.3 Encoding automaton of a bipolar code.

In this case, completely different from the unipolar code, two digits regarded as stochastic variables do not necessarily become independent even if they are in the different blocks. Consequently, $\overline{R}(k)$ does not always become zero even if k is sufficiently large.

In order to evaluate a line spectrum, a local average of a bipolar coded error control code with block length N is firstly considered. Let us think of a state transition diagram observed at every N time units in Fig. 6.3, assuming that s_1 is the initial state. The diagram showing state transition among the successive first digits of each block is shown in Fig.6.4, with only transition probabilities being attached to each state transition.

Accordingly, if the condition

$$0 < \Pr(c_1 \oplus \ldots \oplus c_N = 1) < 1$$

is met, then the steady state occupancy probabilities for s_1 and s_2 are both 0.5, which implies that an encoding

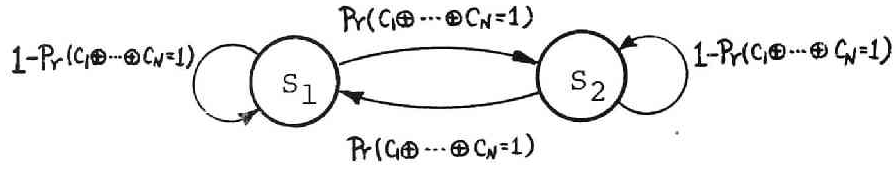


Fig.6.4 State transition diagram observed at every N time units (N: block length).

automaton occupies the states s_1 and s_2 with equal probability at the beginning of every blocks. On the other hand, apparently from Fig.6.3, the output sequences for the same input sequence started from s_1 and s_2 are the negative of each other. From these facts, we can conclude that all local averages must be zero, i.e., $m_1=m_2=\ldots=m_N=0$, in the case of $0 < \Pr(c_1 \oplus \ldots \oplus c_N=1) < 1$.

In other cases where $\Pr(c_1 \oplus \ldots \oplus c_N=1)$ equals to 0 or 1, however, the circumstances become fairly different from the previous case. The state transition diagrams among the successive first digits of the block can be depicted in Fig.6.5 and Fig.6.6.

Fig.6.5 $\Pr(c_1 \oplus \ldots \oplus c_N=1)=0$.

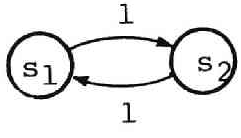


Fig.6.6 $\Pr(c_1 \oplus \ldots \oplus c_N=1)=1$.

In the case of Fig.6.5, $c_1 \oplus \ldots \oplus c_N$ always equals to 0, which corresponds to the case where an even parity check bit is added to the block. If the s_1 is selected as the initial state, then generally m_1 becomes positive, whereas m_N becomes negative. This guarantees the existence of a line spectrum. In the case of Fig.6.6, it is thought to be the case where an odd parity check bit is attached

to every block. The period of a state transition diagram is $2N$ generally, and $m_1>0$, $m_N>0$, $m_{N+1}<0$ and $m_{2N}<0$ hold true, which assures the presence of a discrete line spectrum. By summarizing the above discussion, we obtain the following proposition.

[Proposition 6.3] A discrete line spectrum of a bipolar coded error control code surely exists only when the equation $c_1 \oplus \ldots \oplus c_N=0$ or 1 holds true for every block, as exemplified by a usual parity check code or an extended Hamming code, and never appears otherwise.

With regard to the actual calculation method of a line spectrum for a parity check code, it will be illustrated using an example later.

Next, a continuous spectrum should be considered. To calculate it, it is necessary to compute autocovariances $\bar{R}(k)$ for any k. The above discussion suggests, however, that two output digits in separate blocks become statistically independent in the case of $c_1 \oplus \ldots \oplus c_N=0$ or 1, because the initial state of any block is uniquely determined independently of the past outputs. This implies $\overline{R}(k)=0$ for any $k \geq N$. Therefore, only the calculation of $\overline{R}(0),\ldots,$ $\overline{R}(N-1)$ is required.

On the other hand, in the case of $0 < \Pr(c_1 \oplus \ldots \oplus c_N=1) < 1$, $\overline{R}(k) \neq 0$ holds true even for sufficiently large k. However, because of the fact that $m_i=0$ for $1 \leq i \leq N$, $\overline{R}(k)$ agrees with $R(k)$ for any k. As a first step to compute $R(k)$, let us figure out $r_i(k)= \mathrm{av}(a_i a_{i+k})$ (i: fixed) assuming that a_i and a_{i+k} belong to the separate blocks. As shown in Fig.6.7, let a_i and a_{i+k} be separated by n blocks and c_i and c_j correspond to a_i and a_{i+k}, respectively ($j=i+k \bmod N$). In this case, it follows from the bi-

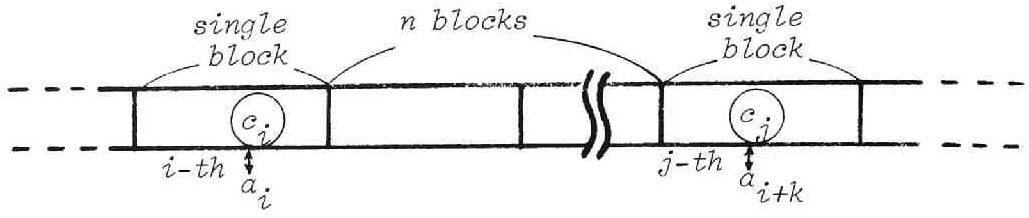


Fig.6.7 Block coded sequence where c_i and c_j are separated by n blocks.

polar encoding law that a_i and a_{i+k} have the same polarity pulses if mod 2 sum of intermediate input symbols is 1 and the reverse polarity if 0. Consequently,

$$r_i(k)=\mathrm{av}(a_i a_{i+k})=\Pr(c_i=1\wedge c_{i+1}\oplus\ldots\oplus c_{j-1}=1\wedge c_j=1)$$
$$-\Pr(c_i=1\wedge c_{i+1}\oplus\ldots\oplus c_{j-1}=0\wedge c_j=1)$$
$$=2\Pr(c_i=1\wedge c_{i+1}\oplus\ldots\oplus c_{j-1}=1\wedge c_j=1)-\Pr(c_i=1\wedge c_j=1). \quad (6.10)$$

Here, two kinds of joint probabilities within the same block are defined as follows:

$$\alpha_i \triangleq \Pr(c_i^{(\gamma)}=1 \wedge c_{i+1}^{(\gamma)}\oplus\ldots\oplus c_N^{(\gamma)}=1) ,$$

$$\beta_i \triangleq \Pr(c_1^{(\gamma)}\oplus\ldots\oplus c_{i-1}^{(\gamma)}=1 \wedge c_i^{(\gamma)}=1) .$$

In addition, if we let the probability that mod 2 sum of all the digits over whole n blocks, $\Pr\left(\sum_{\gamma=1}^{i+n-1}\sum_{i=1}^{N} c_i^{(\gamma)}=1\right)$, be g_n, then

$$g_n = (1-(1-2g_1)^n)/2 ,$$

where $g_1 = \Pr(c_1^{(\gamma)}\oplus\ldots\oplus c_N^{(\gamma)}=1)$ and is readily evaluated by use of Lemma 6.1. Then, by noting the fact that

$$\Pr(c_i^{(\gamma)}=1 \wedge c_{i+1}^{(\gamma)}\oplus\ldots\oplus c_N^{(\gamma)}=0) = p_i-\alpha_i ,$$

$$\Pr(c_1^{(\gamma+n+1)}\oplus\ldots\oplus c_{j-1}^{(\gamma+n+1)}=0 \wedge c_j^{(\gamma+n+1)}=1)=p_j-\beta_j ,$$

we obtain

$$\Pr(c_i=1 \wedge c_{i+1}\oplus\ldots\oplus c_{j-1}=1 \wedge c_j=1)$$

$$=\Pr(c_i^{(\gamma)}=1 \wedge (c_{i+1}^{(\gamma)}\oplus\ldots\oplus c_N^{(\gamma)})\oplus(c_1^{(\gamma+1)}\oplus\ldots\oplus c_N^{(\gamma+\eta)})\oplus(c_1^{(\gamma+\eta+1)}\oplus\ldots\oplus c_{j-1}^{(\gamma+n+1)})=1 \wedge c_j^{(\gamma+n+1)}=1)$$

$$=\alpha_i(1-g_n)(p_j-\beta_j)+(p_i-\alpha_i)g_n(p_j-\beta_j)+(p_i-\alpha_i)(1-g_n)\beta_j+\alpha_i g_n\beta_j$$

$$=p_i p_j\ /2\ -(p_i-2\alpha_i)(p_j-2\beta_j)(1-2g_1)^n/2\ .$$

Accordingly,

$$r_i(k)\ =\ -(p_i-2\alpha_i)(p_j-2\beta_j)(1-2g_1)^n. \qquad (6.11)$$

If we let the Hamming weight of a vector $\mathbf{h}_1\oplus\mathbf{h}_2\oplus\ldots\oplus\mathbf{h}_N$ be h, where $\mathbf{a}\oplus\mathbf{b}$ means the bitwise mod 2 addition of vectors $\mathbf{a}$ and $\mathbf{b}$, then $g_1 = (1-(1-2p)^h)/2$, so we obtain the next proposition.

[Proposition 6.4] The relation

$$R(k+rN)\ =\ R(k)\ (1-2p)^{rh} \qquad (6.12)$$

generally holds true for $k \geq N$ for any bipolar coded error control code satisfying the condition $0 < \Pr(c_1\oplus\ldots\oplus c_N=1) < 1$. It follows that a normalized power spectrum can be represented in terms of $R(0)$, $R(1)$, ... , $R(2N-1)$ only as

$$S(f)\ =\ R(0)+2\sum_{i=1}^{N-1}R(i)\cos i\omega T+$$

$$+2\sum_{i=N}^{2N-1}R(i)\ \frac{\cos i\omega T\ -(1-2p)^h\cos(N-i)\omega T}{1-2(1-2p)^h\cos N\omega T\ +(1-2p)^{2h}} \qquad (6.13)$$

where N is the block length and h is the Hamming weight of the vector $\mathbf{h}_1\oplus\mathbf{h}_2\oplus\ldots\oplus\mathbf{h}_N$.

(Proof) Eq.(6.12) can be derived directly from Eq.(6.11).

Eq.(6.13) is obtained by the following procedures:

$$S(f) = R(0)+2\sum_{i=1}^{\infty} R(i)\cos i\omega T$$

$$=R(0)+2\sum_{i=1}^{N-1} R(i)\cos i\omega T+2\sum_{i=N}^{2N-1}\sum_{r=0}^{\infty} R(i+rN)\cos(i+rN)\omega T.$$

Here, by letting the third term of the above expression be W and substituting Eq.(6.12) to it, we have

$$W = 2\sum_{i=N}^{2N-1} R(i)\sum_{r=0}^{\infty}(1-2p)^{rh}\cos(i+rN)\omega T$$

$$=2\sum_{i=N}^{2N-1} R(i)\sum_{r=0}^{\infty}(1-2p)^{rh}\frac{e^{j(i+rN)\omega T}+e^{-j(i+rN)\omega T}}{2}$$

$$=2\sum_{i=N}^{2N-1} R(i)\frac{1}{2}\left(\frac{e^{ji\omega T}}{1-(1-2p)^{h}e^{jN\omega T}}+\frac{e^{-ji\omega T}}{1-(1-2p)^{h}e^{-jN\omega T}}\right)$$

$$=2\sum_{i=N}^{2N-1} R(i)\frac{\cos i\omega T-(1-2p)^{h}\cos(N-i)\omega T}{1-2(1-2p)^{h}\cos N\omega T+(1-2p)^{2h}}\ .$$

Thus, we have proven the Proposition 6.4.

From the above proposition, it is necessary to know totally $2N$ values, namely $R(0)$, $R(1)$, ... , $R(2N-1)$, in order to calculate continuous power spectrum. Among them, $R(N)$, ... , $R(2N-1)$ are readily calculated by Eq.(6.11), i.e.,

$$R(k)=-(1/N)\sum_{i=1}^{N}(p_i-2\alpha_i)(p_{\substack{i+k\\(mod N)}}-2\beta_{\substack{i+k\\(mod N)}})(1-2p)^{h\eta(i,k)} \qquad (N<k<2N) \tag{6.14}$$

where $\eta(i,k)$ is the number of intermediate blocks between a_i and a_{i+k}. As to p_i, α_i and β_i, they are all obtained systematically by using Prop.6.1 and 6.2. So the only unknown quantities left to be calculated are $R(0)$, ... ,

$R(N\text{-}1)$. However, if we take Eq.(6.11) into account, we can restrict the calculation of $R(k)$ $(0 \leq k \leq N\text{-}1)$ to that of $r_i(k)$ under the condition that a_i and a_{i+k} belong to the same block. This fact, coupled with Eq. (6.10), means that it is the joint probability

$$\Pr(c_i=1 \wedge c_{i+1} \oplus \ldots \oplus c_{j-1}=1 \wedge c_j=1)$$

under the condition that c_i and c_j are within the same block that we really need to evaluate.

— *Calculation of a joint probability* $\Pr(c_i=1 \wedge c_{i+1} \oplus \ldots \oplus c_{j-1}=1 \wedge c_j=1)$ *under the assumption that both* c_i *and* c_j *belong to the same block* —

It seems rather complex to calculate the above joint probability when compared with $\Pr(c_i=1 \wedge c_j=1)$ under the same condition. However, if we make use of a minterm expression of a logic function again, we can find a systematic procedure. To simplify the minterm expression, information digits $c_1, \ldots, c_K$ are divided into disjoint seven classes, which are in turn taken as the independent variables for the minterm expression. Division of information digits is carried out according to whether a given information digit appears in vectors $\mathbb{h}_i$, $\mathbb{h}_{i+1} \oplus \ldots \oplus \mathbb{h}_{j-1}$, or $\mathbb{h}_j$. As a result, totally seven possibilities or classes arise except for the case where a digit appears none of three vectors, and they are shown in Fig.6.8 as A(1) through A(7). It should be noted that only seven classes will always do whatever large the number of information digits K may be. In Fig.6.8, A(1) is, for example, a set of information digits which are contained only in $\mathbb{h}_i$ and not in neither $\mathbb{h}_{i+1} \oplus \ldots \oplus \mathbb{h}_{j-1}$ nor $\mathbb{h}_j$, whereas A(3) is a set of information digits which are included both in $\mathbb{h}_i$ and $\mathbb{h}_{i+1} \oplus \ldots \oplus \mathbb{h}_{j-1}$ and not in $\mathbb{h}_j$.

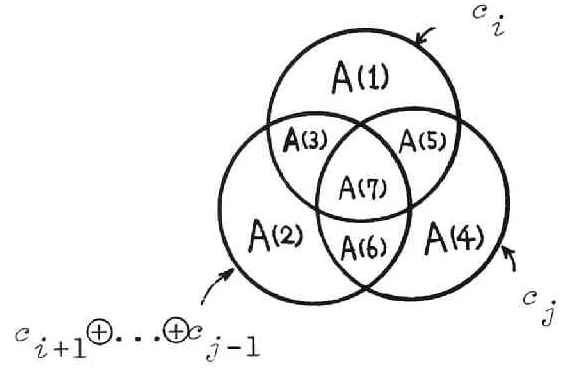


Fig.6.8 Partition of information digits.

If we denote the mod 2 sum of all the digits contained in A(i) by A'(i) for $1 \leq i \leq 7$, then we have

$$\Pr(c_i=1 \wedge c_{i+1} \oplus \ldots \oplus c_{j-1}=1 \wedge c_j=1)$$
$$=\Pr(A'(1) \oplus A'(3) \oplus A'(5) \oplus A'(7) \wedge A'(2) \oplus A'(3) \oplus$$
$$A'(6) \oplus A'(7) \wedge A'(4) \oplus A'(5) \oplus A'(6) \oplus A'(7)=1) \quad (6.15)$$

To evaluate the above probability, we must check out all the conditions of A'(1) through A'(7) such that the condition stated within the parenthesis becomes true. This is equivalent to obtaining a minterm expression for a Boolean function $A'(1) \oplus A'(3) \oplus A'(5) \oplus A'(7) \wedge A'(2) \oplus A'(3) \oplus A'(6) \oplus A'(7) \wedge A'(4) \oplus A'(5) \oplus A'(6) \oplus A'(7)$. Once the minterm expression is obtained, the probability is readily calculated by simply substituting $\Pr(A'(i)=1)$ and $\Pr(A'(i)=0)$ to the variables A'(i) and $\overline{A'(i)}$ respectively. In this case, totally 16 minterms appear because seven variables are subject to constraint by three independent equations. We shall show them in the Table 6.1. Each row corresponds to a single minterm. For example, 1101000 corresponds to $A'(1)A'(2)\overline{A'(3)}A'(4)\overline{A'(5)}\,\overline{A'(6)}\,\overline{A'(7)}$. On the other hand,

if we let the cardinality of A(i) be n_i, then it is apparent that

$$\Pr(A'(i)=1)=f(n_i),$$

$$\Pr(A'(i)=0)=1-f(n_i).$$

Summarizing the above discussion, we can obtain the following proposition.

[Proposition 6.5] A joint probability $\Pr(c_i=1 \wedge c_{i+1} \oplus \cdots \oplus c_{j-1}=1 \wedge c_j=1)$, under the condition that c_i and c_j belong to the same block, is computed by the following formula:

$$\Pr(c_i=1 \wedge c_{i+1} \oplus \cdots \oplus c_{j-1}=1 \wedge c_j=1) = \sum_{i=1}^{16} \prod_{j=1}^{7} [E(i,j)f(n_j)+(1-E(i,j))(1-f(n_j))],$$

where n_i is the number of elements in A(i) and E(i,j) is the (i,j) entry of a two dimensional array shown in Table 6.1 each row of which corresponds to a minterm of the logic function discussed above.

Using above proposition and Eq.(6.11), $R(1),\ldots,R(N-1)$ are easily calculated, while

$$R(0)=\frac{1}{N}\sum_{i=1}^{N} \mathrm{av}(a_i^2)=\frac{1}{N}\sum_{i=1}^{N} p_i .$$

By introducing two matrices $R_1 \triangleq (r_{ij}^{(1)})$ and $R_2 \triangleq (r_{ij}^{(2)})$, the procedure for evaluating $R(0)$,...,

Table 6.1

Array E(16,7) representing set of minterms making logical function $A'(1)\oplus A'(3)\oplus A'(5)\oplus A'(7) \wedge A'(2)\oplus A'(3)\oplus A'(6)\oplus A'(7) \wedge A'(4)\oplus A'(5)\oplus A'(6)\oplus A'(7)$ true.

$A'(1)$	$A'(2)$	$A'(3)$	$A'(4)$	$A'(5)$	$A'(6)$	$A'(7)$
1	1	0	1	0	0	0
0	0	0	0	0	0	1
1	0	0	0	0	1	0
0	1	0	1	0	1	1
0	1	0	0	1	0	0
1	0	0	1	1	0	1
0	0	0	1	1	1	0
1	1	0	0	1	1	1
0	0	1	1	0	0	0
1	1	1	0	0	0	1
0	1	1	0	0	1	0
1	0	1	1	0	1	1
1	0	1	0	1	0	0
0	1	1	1	1	0	1
1	1	1	1	1	1	0
0	0	1	0	1	1	1

$R(2N-1)$ is restated more systematically. Namely let $r_{ij}^{(1)}$ and $r_{ij}^{(2)}$ be defined as follows:

$$r_{ij}^{(1)} = \begin{cases} \Pr(c_i=1 \wedge c_{i+1} \oplus \cdots \oplus c_{j-1}=1 \wedge c_j=1) & \text{given in Prop.6.5, } (i<j) \\ p_i & (i=j), \\ -(p_i-2\alpha_i)(p_j-2\beta_j) & (i>j), \end{cases}$$

$$r_{ij}^{(2)} = \begin{cases} -(p_i-2\alpha_i)(p_j-2\beta_j) & (i \leq j), \\ -(p_i-2\alpha_i)(p_j-2\beta_j)(1-2g_1) & (i>j). \end{cases}$$

Then, $R(0), \ldots, R(2N-1)$ are readily obtained by taking summations along the paths, schematically drawn in Fig. 6.9, and then dividing them by N.

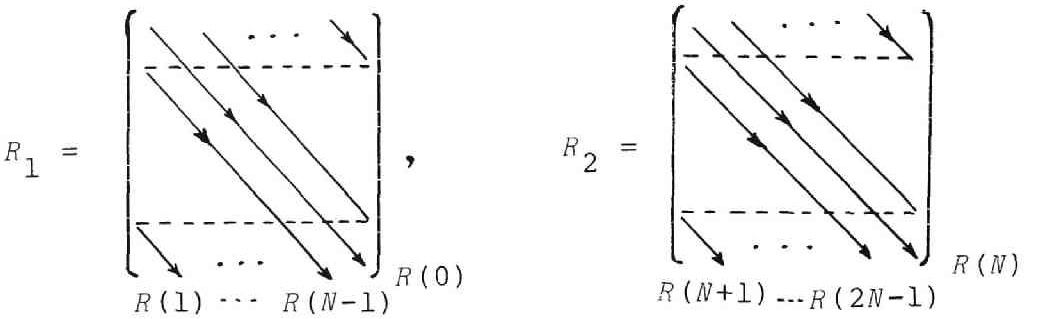


Fig. 6.9 Matrices with arrows depicted to show calculation procedure for $R(0), \ldots, R(2N-1)$.

More strictly speaking, they are calculated by

$$R(k) = \frac{1}{N}\left(\sum_{i=1}^{N-k} r_{i,i+k}^{(1)} + \sum_{j=1}^{k} r_{N-k+j,j}^{(1)}\right) \qquad (0 \leq k \leq N-1),$$

$$R(k) = \frac{1}{N}\left(\sum_{i=1}^{N-k} r_{i,i+k}^{(2)} + \sum_{j=1}^{k} r_{N-k+j,j}^{(2)}\right) \qquad (N \leq k \leq 2N-1).$$

These results coupled with Prop.6.4 give our goal, i.e.

power spectrum of a bipolar coded error control code.

As examples, the above procedure is applied to evaluate power spectra of bipolar coded (7,4) Hamming code, (15,11) Hamming code and (15,7) double error correcting code assuming that p=0.3. The computed results together with a usual bipolar code power spectrum are depicted in Fig.6.10. The figure is showing that the power will increase for error control codes if compared with normal bipolar sequence. This is due to the fact that check digits generally represented as mod 2 sum of information digits have higher probabilities of being ±1 because $f(n) > f(1)=p \quad (n > 2)$ holds true for $p < 0.5$. However, if $p > 0.5$, the circumstances will make a complete change and power for error control codes will decreases since $f(n) < f(1)=p \quad (n \geq 2)$. We also notice that (15,7) code is rather oscillatory in its power spectrum.

In the case of $0 < Pr(c_1 \oplus \ldots \oplus c_N=1) < 1$, we can show that power spectra of bipolar coded error control codes completely coincide with that of normal sequence if $p=0.5$. It is verified from the fact that $p_i=0.5$, $\alpha_i=\beta_i=0.25$ $(1 \leq i \leq N)$ and $g_1=0.5$ if $p=0.5$.

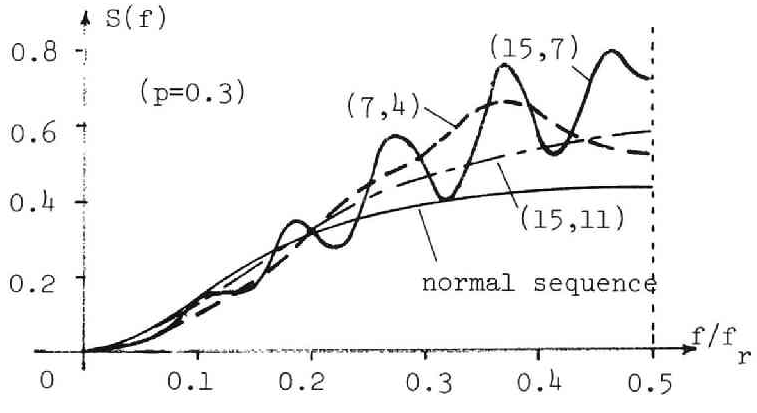


Fig.6.10 Normalized power spectra $S(f)$ of bipolar coded error control codes.

In other case where $Pr(c_1 \oplus \ldots \oplus c_N = 1) = 0$ or 1, however, discrete line spectra will appear even when $p=0.5$ and, as a result, continuous component will decrease.

[Proposition 6.6] The continuous power spectrum of bipolar coded $(N, N-1)$ parity check code, $w_c^{(N)}(f)$, becomes

$$w_c^{(N)}(f) = \frac{N-1}{N} w_c(f)$$

if $p=0.5$, where $w_c(f)$ is the continuous power spectrum of normal bipolar sequence and written as

$$w_c(f) = \frac{1}{2} f_r |G(f)|^2 (1-\cos\omega T) .$$

On the other hand, the discrete line spectrum, $w_d^{(N)}(f)$, becomes

$$w_d^{(N)}(f) = \frac{1}{2N^2} f_r^2 |G(f)|^2 \sum_{k=-\infty}^{\infty} (1-\cos 2\omega T)\, \delta_d(f - \frac{k}{N} f_r) .$$

Apparently,

$$\int_0^{f_r} w_c(f)\, df = \int_0^{f_r} (w_c^{(N)}(f) + w_d^{(N)}(f))\, df$$

holds since transmitted power is left unchanged when $p=0.5$.

(Proof) In Fig. 6.3, let us assume that s_1 is the initial state. Then, in the case of $p=0.5$, we have

$$m_1 = 0.5,\ m_2 = \ldots = m_{N-1} = 0,\ m_N = -0.5,$$

because generally $m_1 = p$, $m_i = p(1-2p)^{i-1}$ $(1 < i < N)$ and $m_N = -(1/2)(1-(1-2p)^{N-1})$. Accordingly, from Eq.(2.14), we obtain line spectrum:

$$w_d^{(N)}(f) = \frac{1}{4N^2} f_r^2 |G(f)|^2 \sum_{k=-\infty}^{\infty} |1-e^{-j2\pi k\frac{1}{N}}|^2 \delta_d(f-\frac{k}{N}f_r)$$
$$= \frac{1}{2N^2} f_r^2 |G(f)|^2 \sum_{k=-\infty}^{\infty} (1-\cos 2\pi\frac{k}{N}) \delta_d(f-\frac{k}{N}f_r) \quad .$$

As for continuous spectrum, autocovariances $R(k)$ for $k \geq 0$ must be evaluated first. However, the condition that $c_1 \oplus \ldots \oplus c_N = 0$ (or 1) holds for every block assures independence between two digits belonging to separate blocks. So, we have

$$\overline{R}(k) = \begin{cases} (1/N) \sum_{i=1}^{N-k} \{av(a_i a_{i+k}) - m_i m_{i+k}\} & (0 \leq k \leq N-1), \\ 0 & (N \leq k). \end{cases}$$

It is straightforward to calculate $\overline{R}(k)$ for $0 \leq k \leq N-1$, if we define two matrices $Z=(z_{ij})$ and $M=(m_{ij})$ where $z_{ij} \triangleq av(a_i a_j)$ and $m_{ij} \triangleq m_i m_j$. In this case, they are

$$Z = \begin{pmatrix} 1/2 & -1/4 & 0 & \cdots & 0 & -1/4 \\ & 1/2 & -1/4 & \ddots & & 0 \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & \ddots & \ddots & 0 \\ & & & & \ddots & -1/4 \\ & & & & & 1/2 \end{pmatrix}, \quad M = \begin{pmatrix} 1/4 & 0 & 0 & \cdots & 0 & -1/4 \\ & 0 & 0 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & 0 & 0 \\ & & & & \ddots & 1/4 \end{pmatrix}.$$

Accordingly, $\overline{R}(0) = \frac{1}{2}\frac{N-1}{N}$, $\overline{R}(1) = -\frac{1}{4}\frac{N-1}{N}$, $\overline{R}(2)=\overline{R}(3)=\ldots=0$. Thus, we have

$$S_c(f) = (\frac{N-1}{N})\frac{1}{2}(1-\cos\omega T) \quad .$$

The last relation in the above proposition indicating the conservation of power is almost apparent, if one takes the fact that $f(1)=f(2)=...=f(N)=0.5$ for $p=0.5$ into account. .

The procedure developed here will be also applied to such transmission codes that have relatively few states in the state transition diagram like duobinary code.

In case of nonlinear error control code, however, check digits are usually not simply linear summations but nonlinear functions of information digits. So, it is impossible to specify the encoding law by means of parity check matrix. As a result, matrix row operation based procedure for evaluating power spectrum is not directly applicable. However, minterm expression based probability calculation method is of more general use and can be applied even for a nonlinear error control code. Namely, when check digit is known to be a nonlinear Boolean function of information digits, we can carry out the probability calculation simply by deriving a minterm expression and substituting occurrence probability of each variable to it, although grouping of information digits to decrease the number of Boolean variables for simplfying minterm expression is mostly not possible.

CHAPTER 7 THE OPTICALLY LINKED LABORATORY COMPUTER NETWORK LABOLINK

Some researches made by the author during the development of a laboratory computer network LABOLINK are described, together with the overall outline of LABOLINK, in this chapter.

7.1 Introduction to LABOLINK

A laboratory computer network that demonstrates an easy yet versatile way of accessing several kinds of computers from a single, small-scale computer has been developed in Professor Yajima's research laboratory at Kyoto University. Named as "LABOLINK", the network is designed to support laboratory research and education. The author had a chance to take part in the LABOLINK project and engaged in the network development with many other members in the laboratory under the supervision of Professor Yajima.

After presenting a general description of LABOLINK in the rest of this section, some network components like a single-line multi-connector and an optical fiber data transmission system which the author worked at directly are stated in the next section, followed by the description of simple encoding scheme of a Fibonacci code used in a single-line multi-connector.

Although several small-scale network configurations have been reported, LABOLINK aims at several new approaches. For example, it seems to be the first working computer network to be linked by fiber optic cables. In addition, LABOLINK required no modification of the existing operating system.

As user requirements, in LABOLINK, two major kinds of requirements had to be satisfied by the network. First, users had to be able to access a variety of computers from any terminal in the laboratory. Moreover, it had to permit high-speed file transfers via available communication lines between remote computers and the central processors of the laboratory. An additional desirable feature was the inclusion of data transmission capability between the remote computers via user-dedicated minicomputers.

The other requirements dealt with design considerations:

(1) The system had to be simple, easy to develop, and comprehensible to the users, inasmuch as development resources would often be restricted and software development would be left largely to the users themselves.

(2) The system had to be modular, expandable, and modifiable.

(3) A systematic interface design procedure had to be employed.

On the basis of these requirements, the following design features were adopted:

(1) A dedicated PDP 11/40 acts as both CPU and communication controller for the high-speed channels used to reach the remote computers/computer networks. Low-speed channels for user terminals are controlled by a microprocessor.

(2) Network supporting software is executed as user programs, which can communicate with each other directly without a network control subsystem within the operating system. To support software development in the dedicated mini, a communication control facility was added to multi-user BASIC [34]. This extended BASIC permits LABOLINK resources to be accessed simultaneously from as many as eight terminals.

In addition, several new experimental approaches to network organization have been adopted in LABOLINK.

An example is the use of optical fiber cables for communication lines. The optical fiber connections were installed primarily to establish broadband communication channels between research facilities and to demonstrate their practical use. Coupled with a kind of universal digital multiplexer, called a single-line multi-connector, they permit several kinds of logical communication lines to be achieved. They also permit any network configuration to be embedded on previously established channels without concern for line capacity.

The single-line multi-connector, which represents yet another unique development, makes it possible to replace the usual multi-wire cable in a nearly plug-compatible fashion using either a pair of optical fiber cables, one for each direction, or a single copper cable. Two types of SLMC's have been developed and are used both in high-speed communication lines for file transfers as well as low-speed lines for terminals.

LABOLINK currently consists of a PDP 11/40 in the laboratory linked with a FACOM 230-75 in the university's Data Processing Center and a HITAC 8350 in the Department of Information Science as shown in Fig.7.1. The PDP 11/40

is linked via 1200-bps line with the FACOM, which is to be a host computer in a nationwide inter-university computer network. This network will permit users to access several large computing resources or large data bases. Also, using a 1-Mbps in-house line liking the PDP 11/40 with the HITAC, the user can share the HITAC's magnetic files and I/O devices. Terminals distributed in research rooms are centralized through low-speed lines and controlled by a microprocessor-based terminal switching system developed by the laboratory.

Interconnection between the PDP 11/40 and the HITAC computer is realized by two hardware interface devices

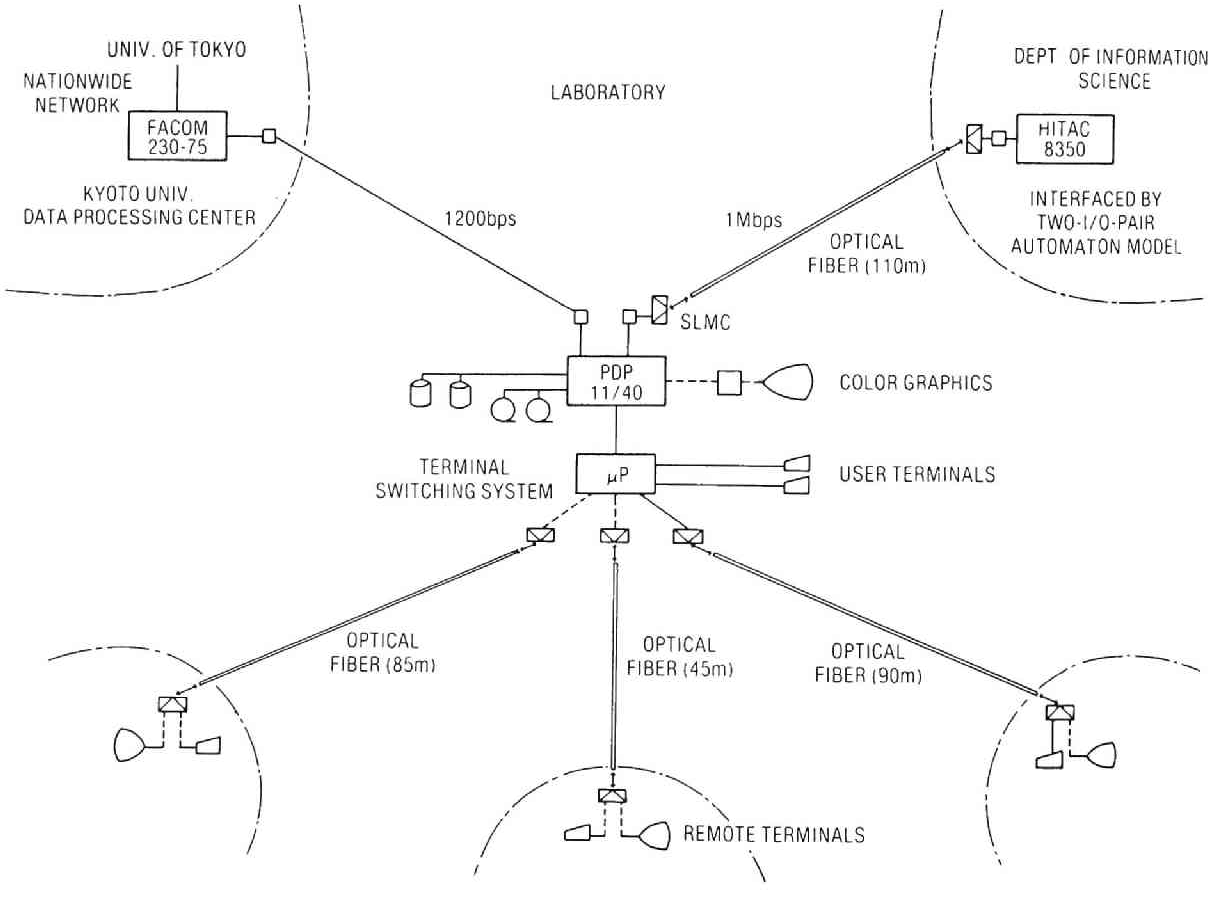


Fig. 7.1 LABOLINK configuration.

which regulate flow control and error control (usually executed by software)[35].

Some hardware components have been developed which serve to simplify software development. One of them, an LSI ROM based code converter between ASCII and EBCDIC codes, attaches directly to the unibus of the PDP 11/40. It is designed so that if one moves a specific character code to be converted to the internal register of the component, then the converted result from the same register can be read out immediately by the next instruction. It can be used not only from assembly language programs but also from extended multi-user BASIC programs.

7.2 Optical Fiber Transmission and Single-Line Multi-Connector

As indicated earlier, optical fiber cables rather than conventional copper wire cables have been adopted as transmission lines for the in-house link of LABOLINK. The optical fiber's broad bandwidth is utilized by coupling them with a new kind of digital multiplexer called a single-line multi-connector(SLMC).

Among the advantages of optical fiber are its extremely low transmission loss, small physical size, and high inductive-noise immunity. In addition, the materials used in its manufacture are in plentiful supply.

The following system characteristics have been realized by utilizing optical fiber cables in the LABOLINK

network:

(1) Broadband communication channels between research rooms have been achieved. Even ultra-high speed image transmission will be possible.

(2) An optical fiber permits the establishment of arbitrary logical communication links by means of multi-purpose time-multiplexing, using the SLMC's connected at both ends.

An optical fiber and its cable structure, as employed in LABOLINK, are shown in Fig. 7.2. A plastic clad multi-mode optical fiber with core diameter of about 200 μm and transmission loss of about 15 dB/km at wavelength 910 nm (near infrared) has been installed. A GaAs LED of wavelength 910 nm and a pin diode are used as a light source and a photo detector. Although 1 mW of power is emitted from an LED, only about 10 μW is input to an optical fiber because of the coupling loss of 20 dB. The outline of the transmission system is illustrated in Fig. 7.3.

When one considers the efficient use of such an optical fiber transmission line, it seems that conventional

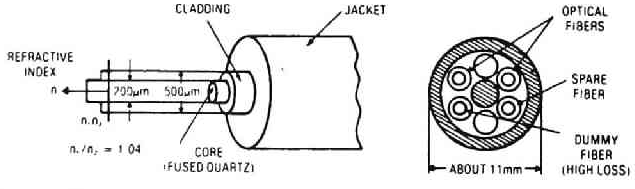


Fig.7.2 Plastics-clad multi-mode optical fiber and its cable structure.

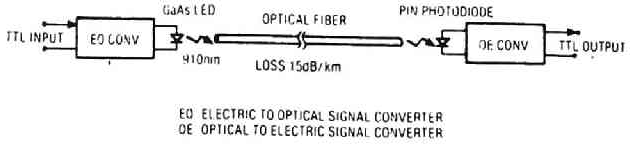


Fig.7.3 Optical fiber transmission system.

multiplexing techniques, primarily designed for homogeneous signal lines, are not sufficient. Hence, a new, general purpose digital multiplexer has been developed suitable for asynchronous heterogeneous signal lines. That is, a multiplexing device has been developed that is especially suitable for broadband optical fiber transmission lines, into which arbitrary combinations of physical communication lines can be embedded. Thus, as previously mentioned, it is possible to replace the usual multi-wire cables with multi-connectors in nearly plug-compatible fashion by a pair of optical fibers (one for each direction) or by a single copper cable (thereby reducing the number of signal lines, and improving reliability and maintainability). Serial rather than parallel transmission is utilized, thereby providing the intrinsic advantage of being completely free from racing problems or skew of the parallel propagation of signals along multi-wires. The multiplexer also differs from the usual data multiplexing LSI(UART or USRT) in that it determines the time for sampling the data to be transmitted in serial. This time determination is made not by externally given strobe signals, but by either detecting the level changes in signal lines, or by sufficiently high speed successive samplings of the parallel lines.

Two kinds of SLMC's have thus far been implemented. The Type I SLMC has been designed on the basis of the device to be connected. All the signal lines are classified into two groups— control signal lines and strobed data lines. The 8-bit data lines are treated in a bundle as if they were a single control line represented by a strobe signal. It is designed on the basis of the pinciple that each level change in the control signal lines is detected to be transmitted over a single line serially.

Thus, it achieves relatively efficient multiplexed data transmission. By employing pulse width modulation with three kinds of pulse widths (250 nsec, 500 nsec, and 750 nsec), the Type I SLMC realizes transmission rate of 1M baud. It is actually applied to the link between the HITAC computer and the PDP 11/40.

In contrast, the Type II SLMC is designed independent of the devices to be connected. By sampling signal lines in a high-speed manner, it carries out serial data transmission. As a result, it is less efficient than Type I, but has more generality. A run of zeros at the tail end of sampled parallel data is automatically suppressed, thus increasing the effective sampling rate. This enables it to be used not only in fewer high-speed channels but also in more lower-speed channels. However, it can be operated by externally applied clock pulses of arbitrary speed (user selectable) and thus is different from the usual parallel to serial converter. A typical line signal used in a Type II SLMC is shown in Fig. 7.4. A block diagram is illustrated in Fig. 7.5. Table 7.1 summarizes specifications for both types of SLMC's.

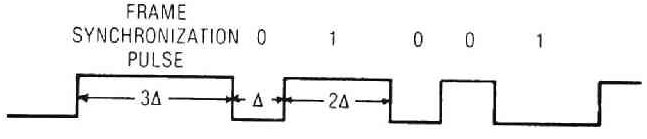
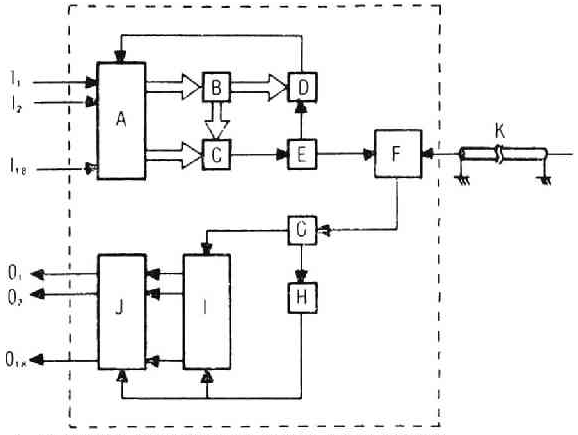


Fig. 7.4 Typical line signal of a single-line multi-connector (type II).



A: PARALLEL-IN/PARALLEL-OUT SHIFT REGISTER
B: PRIORITY ENCODER
C: DATA MULTIPLEXER
D: PRESETTABLE COUNTER
E: TRANSMISSION CODER
F: LINE ADAPTER (SENDING/RECEIVING SIGNAL SEPARATION CIRCUIT)
G: DECODER
H: FRAMING SIGNAL DETECTOR
I: PARALLEL OUTPUT SHIFT REGISTER
J: DATA LATCHES
K: A PAIR OF OPTICAL FIBERS (OR A SINGLE COPPER WIRE)

Fig. 7.5

Block diagram of a single-line multi-connector (Type II).

SPECIFICATION	TYPE I	TYPE II
TRANSMISSION RATE	1 Mbps	4 Mbps (max)
MODULATION	PWM	modified PWM (Fibonacci code)
PARALLEL TO SERIAL CONVERSION	SUCCESSIVE DETECTION OF LEVEL CHANGES	SUCCESSIVE HIGH SPEED SAMPLING OF PARALLEL LINES
SOME FEATURES	EFFICIENT TRANSMISSION BY USE OF SEMANTICS OF SIGNALS	AUTOMATIC SUPPRESSION OF RUN OF ZEROS AT THE TAIL END OF BLOCK
	TWO CLASSES OF SIGNAL LINES (STROBED DATA SIGNAL AND CONTROL SIGNALS)	NOTHING SENT WHEN ALL THE LINES ARE IN NORMAL STATES
		LSI ORIENTED DESIGN
	ERROR CONTROL WITH RESPECT TO SIGNAL LINES	CASCADABILITY
NUMBER OF REQUIRED DIGITAL IC's	87(TTL SSI)	44 (TTL SSI, MSI)
POWER DISSIPATION	5 W	5 W
TYPICAL DISTANCE	A FEW HUNDRED METERS	

Table 7.1

Specifications of two kinds of single-line multi-connectors.

7.3 Fibonacci Codes

In type II SLMC, a kind of modified PWM (pulse width modulation) is adopted for serial data transmission as illustrated in Fig. 7.4. It is not the pulse width but the time duration between successive level transitions that convey the information. Consequently, more efficient data transmission can be achieved than simple PWM. Although three kinds of time durations Δ, 2Δ (Δ: time quantum) for binary information and 3Δ for frame synchronization are used, we shall neglect the largest duration 3Δ in this section to focus our discussion on the binary data encoding. Then, such a code can be viewed as a kind of Fibonacci code [36-38].

Fibonacci codes have been thus far proposed as transmission codes having self-synchronization capability. The name Fibonacci code is due to the fact that the encoding scheme is closely related with the well-known Fibonacci series. Roughly speaking, it is encoded by carrying out successive comparisons between Fibonacci series and the number to be coded, each time followed by additions and subtractions. In this section, different from such an encoding method, we will present a very simple encoding algorithm which attains almost the same rate by a substantially simpler circuit.

First, we shall calculate the number of possible codewords of Fibonacci code during the time interval of $n\Delta$. Let it be u_n. Then, we have the recurrence formula:

$$u_n = u_{n-1} + u_{n-2} \quad (n \geq 3),$$

where $u_1=1$ and $u_2=2$, because the last symbol of Fibonacci code is either Δ or 2Δ. By solving this equation, we obtain

$$u_n = (\alpha^{n+1} - \beta^{n+1}) / \sqrt{5},$$

where $\alpha = (1+\sqrt{5})/2$ and $\beta = (1-\sqrt{5})/2$.
Next, we let the minimum time interval necessary to encode all of the k-bit binary sequences in a Fibonacci code, normalized by Δ, be T_k. That is

$$T_k = \min \{ n \mid u_n \geq 2^k \} \quad .$$

In what follows, we shall present a new simple algorithm for Fibonacci encoding and examine its transmission efficiency by comparing it with this T_k for the same k.

Now, we shall describe the encoding algorithm. It is based on the simple direct correspondence between binary sybols 0 and 1 and time durations Δ and 2Δ, different from the conventional algorithm using Fibonacci series. In the case of fixed correspondence, however, the codeword length varies according to a given binary sequence to large extent. Actually, the length of the longest codeword is just twice of that of the shortest codeword. To overcome this inefficiency, it seems to be promising to change the correspondence between binary symbols and time durations adaptively among every data block. That is, shorter time duration is assigned to represent the binary symbol appearing more frequently, and longer one is assigned to represent the binary symbol appearing less frequently in a given sequence. In order to distinguish the two possibilities arising from the adaptive assignment, a mode selection index bit denoting which one is selected must be added to each codeword. In short, we wil formalize the encoding rule as shown in Table 7.2. There, n_1 means the total number of binary symbol 1 appearing in the binary sequence of length k. If $n_1 \leq [k/2]$, for instance, time duration Δ is assigned to a mode selection index bit,

Table 7.2 A simple encoding rule of Fibonacci code.

number of "1"	mode selection index bit	binary symbol "0"	binary symbol "1"
$n_1 \leq [k/2]$	Δ	Δ	2Δ
$n_1 > [k/2]$	2Δ	2Δ	Δ

usually preceding the data sequence, and time durations Δ and 2Δ are assigned to binary symbols 0 and 1, respectively.

As expected, the codeword length still varies in this method according to the contents of the binary sequence. Thus, we define the longest codeword length among the possible binary sequences of fixed length as the codeword length of such a sequence. For a shorter codeword sequence, dummy sequence, consisting of consecutive Δ's, may be added.

Let the codeword length of binary sequences of length k including mode selection index bit be V_k. Then, from Table 7.2, it is easily verified that

$$V_k = \begin{cases} [(3/2)k+1]\Delta & (k\text{:even}), \\ [(3/2)k+3/2]\Delta & (k\text{: odd}). \end{cases}$$

This means we can shorten that codeword length by $([k/2]-1)\Delta$ when compared with the codeword length F_k in case of fixed assignment rule. The value V_k can be shown nearly equal to T_k. Computed values of T_k, V_k and F_k for the same k ranging from 1 to 18 are listed in Table 7.3. So far as this table is concerned, the difference between V_k and T_k is at most Δ, and in some cases they are completely the same. To investigate the limiting behavior of

V_k, we shall calculate the ratio of V_k to T_k when k approaches infinity, i.e.,

$$\lim_{k\to\infty} V_k/T_k = \frac{3}{2}\log_2\frac{1+\sqrt{5}}{2} \fallingdotseq 1.041 \quad .$$

For the sake of comparison, let us calculate the ratio of F_k to T_k

$$\lim_{k\to\infty} F_k/T_k = 2\log_2\frac{1+\sqrt{5}}{2} \fallingdotseq 1.389 \quad .$$

From these calculation, we observe that the simple encoding scheme presented here attains almost the same rate as the conventional encoding scheme based on Fibonacci series by a substantially simple circuit.

Table 7.3 Comparison of codeword lengths of Fibonacci code.

k	T_k	V_k	F_k	k	T_k	V_k	F_k
1	2	3	2	10	16	16	20
2	4	4	4	11	17	18	22
3	5	6	6	12	18	19	24
4	7	7	8	13	20	21	26
5	8	9	10	14	21	22	28
6	10	10	12	15	23	24	30
7	11	12	14	16	24	25	32
8	13	13	16	17	26	27	34
9	14	15	18	18	27	28	36

CHAPTER 8 CONCLUDING REMARKS

In this thesis, digital coded signals have been studied in terms of their power spectral characteristics. This study will be expected to contribute to wide area of baseband digital data transmission systems, such as high speed PCM communication system and digital links among computers and peripheral devices. The discussion about the relation between the structure of encoding automata and the power spectral property can be also applied to multi-level codes generally, and thus will become important in a large capacity digital data transmission system in the near future.

Main results obtained in this study are summarized as follows.

In Chapter 2, a convenient power spectral calculation formula directly applicable to Mealy-type encoding automaton was described. It requires less amount of computation and yet be simpler than already known calculation formula applicable to Moore-type automaton. It was actually applied to calculate power spectra of pulse sequences generated by encoding automata with restricted inputs.

In Chapter 3, a necessary and sufficient condition for a transmission code to have a power spectral null at frequency $f=(h/k)f_r$ is obtained in terms of the structure of encoding automaton. The condition can be checked simply by examining the output symbols along the closed paths up to some specific length. It is a proper generalization of the result that a loop-sum-zero state transi-

tion diagram generates pulse sequences without *dc* component. It will be sure to become a useful guideline toward a systematic construction of an encoding automaton with the specific property.

In Chapter 4, based on the result in Chapter 3, a brief catalog of pseudoternary transmission codes with spectral null at f=0 or $(1/2)f_r$ is obtained in the form of Mealy-type state transition diagram with up to four states. Several new codes are included in a catalog. It seems to have significant practical importance in that it exhaustively lists all possible transmission codes under the specified constraint. A consideration is also made on the derivation of encoding automata with larger states from those with fewer states.

In Chapter 5, we have shown that power spectra of various transmission codes can be reshaped by only imposing a certain restriction on the input sequences. particularly, generation of power spectral nulls or discrete line spectral component is shown to be possible by slightly changing the statistics of input sequences. Both the calculation formula for limiting transmission efficiency and the encoding rule attaining the calculated limit are also developed under a certain input restriction.

In Chapter 6, power spectrum calculation procedure for error control codes such as a multiple error correcting cyclic code is developed. Since an error control code generally requires too many states to represent it by a finite automaton, it is not suitable in this case to apply the result in Chapter 2 directly. So, the new systematic procedure primarily based on the row operations of the parity check matrix corresponding to an error control code is developed. It is actually applied to evaluate the power spectrum of a bipolar coded error

control code, which has never been calculated before, as well as that of unipolar coded one.

In Chapter 7, an optically linked laboratory computer network LABOLINK developed at Professor Yajima's research laboratory was described. It has been constructed by introducing the state-of-the-art technologies and several new approaches. Namely, optical fibers are firstly applied to the communication lines in LABOLINK in order to show their usefulness in digital computer systems. They will be sure to be used in the wide area ranging from small digital systems to large computer networks. An SLMC is another new approach designed to multiplex multi-wire cables. It makes it possible to replace any multi-wire cables by a single wire or a pair of optical fibers in a plug for plug compatible fashion. Fibonacci code which an SLMC type II adopted makes bit and frame synchronizations easy and yet achieves efficient data transmission.

In spite of these research efforts, many problems are left for the future study. Some of them are listed below.

Although a relation between an encoding automaton and its power spectrum was clarified with respect to spectral nulls, almost nothing is known concerning the more general relation between the structure of encoding automata and other important spectral characteristics. For example, the condition that what kind of encoding automaton has sufficient easily extractable timing information seems rather difficult to make clear. Because timing information is usually extracted by performing rectification, a non-linear operation, as in the case of bipolar code.

A catalog construction of wider range of transmission

codes is also desired. A state-level-number based approach seems to be promising with respect to the systematic construction of transmission codes with spectral null at *dc* (f=0).

Spectral reshapings by restricting inputs are other interesting topic to be studied furthermore. A slight deviation of input statistics may possibly bring about unexpected improved spectral characteristics.

Calculation of power spectra of error control codes is also important because error control is becoming rather common in recent data transmission systems. Calculation method developed here can be applied only to encoding automaton with fewer states. So, the development of a more general, yet simple formula is desired.

With respect to LABOLINK, the further development of network and the improvement of an SLMC are expected. Though there exist two types of SLMCs, they have their respective merits and demerits for each application. Thus, further improvement in performance may be possible.

ACKNOWLEDGMENT

The author would like to express his sincere thanks to Professor Shuzo Yajima of Kyoto University for the supervision and continuous encouragement to complete this thesis. Not only he guided the author to the present study but also gave him many helpful suggestions and constructive comments during the course of this study. He also critically read the manuscript of this thesis and gave the author accurate comments to be sincerely appreciated.

The author also wishes to thank Professor Fumio Ikegami for his encouragement and support of this work.

The author is much indebted to Associate Professor Yahiko Kambayashi for his many helpful suggestions and comments.

The author is grateful to Research Associate Hiromi Hiraishi for his useful suggestions in developing computer programs using PDP 11/40, and to the Research Associate Kosaku Inagaki and the other members of Professor Yajima's research group for their useful discussions with the author. Especially, Mr. Kazuo Iwama helped the author in many respects during the network development. Mr. Isao Kamae helped the author very much by performing actual calculation for catalog making in Chapter 4. Concerning Chapter 7, Mr. Katsumi Tanaka and Mr. Tsutomu Nakagawa helped the author in designing SLMC's, whereas the installation of optical fiber transmission lines is due to the personnel in Sumitomo Electric Industries, Ltd.

REFERENCES

[1] A.Croisier, "Introduction to Pseudoternary Transmission Codes," *IBM J. Res. Develop.*, Vol.14, July 1970.

[2] H.Kobayashi, "A Survey of Coding Schemes for Transmission or Recording of Digital Data," *IEEE Trans.*, Vol. COM-19, No.6, Dec. 1971.

[3] M.R.Aaron, "PCM Transmission in the Exchange Plant," *B.S.T.J.*, Vol.41, Jan. 1962.

[4] T.Matsuoka and H.Miyakawa,"Binary-to-Ternary Code Conversion and Intersymbol Interference," *Papers of Technical Group on Information Theory, IECE Japan*, Dec. 1966.

[5] J.M.Sipress, "A New Class of Selected Ternary Pulse Transmission Plans for Digital Transmission Lines," *IEEE Trans.*, Vol.COM-13, Sept. 1965.

[6] A.Lender, "Correlative Level Coding for Binary-Data Transmission," *IEEE Spectrum*, Vol.3, Feb. 1966.

[7] _____, "Correlative Digital Communication Techniques," *IEEE Trans.*, Vol. COM-12, Dec. 1964.

[8] E.R.Kretzmer,"Generalization of a Techniques for Binary Data Communication," *IEEE Trans.*, Vol.COM-14, Feb.1966.

[9] F.Kanaya,"Automaton Model of Digital Transmission Line and Its Application to Logical Code Conversion," *Trans. IECE Japan*, Vol.52-A, Dec.1969.

[10] H.Miyakawa and H.Harashima," A Method of Code Conversion for Digital Communication Channel with Intersymbol Interference," *Trans. IECE Japan*, Vol.52-A, Apr.1969.

[11] Y.Kambayashi and S.Yajima," Properties of Machine Mappings and Their Applications to a Channel with Memory,"*Report of Data Processing Center*, Kyoto University, No.A-8, Jan. 1971.

[12] P.Kabel and S.Pasupathy," Partial-Response Signalling," *IEEE Trans.*, Vol.COM-23, No.9, Sept. 1975.

[13] F.Kanaya,"A Multilevel Transmission Code Generated by Finite Automaton," *Joint Conv. Rec.Four Elec.Inst. Japan*, 2030, 1968.

[14] H.Yasuda and H.Inose, "Direct Calculation Method of

Power Spectrum of Pulse Sequences by Means of Transition Probability Matrices," *Trans. IECE Japan*, Vol.53-A, Nov. 1970.

[15] H.Yasuda," Calculation of Autocorrelation Function and Entropy of Pulse Sequences by Means of Transition Probability Matrices," *Trans. IECE Japan*, Vol.54-A, Sept. 1971.

[16] _____," Calculation of Power Spectra of Pulse Sequences by Means of Transition Probability Matrices," *Trans. IECE Japan*, Vol.54-A, Oct. 1971.

[17] H.Yasuda and H.Inose, "Generalized Method of Constructing Multilevel Balanced Code by Means of Loop-Sum-Zero Transition Diagram,"*Trans.IECE Japan*, Vol.54-A, Sept. 1971.

[18] _____,"A Necessary Condition for Constructing the Balanced Codes,"*Trans.IECE Japan*, Vol.55-A, Apr.1972.

[19] E.Gorog, "Redundant Alphabets with Desirable Frequency Spectrum Properties," *IBM J.Res.Develop.*,Vol.12, May 1968.

[20] G.L.Cariolaro and G.P.Tronca," Spectra of Block Coded Digital Signals,"*IEEE Trans.*, Vol.COM-22, No.10, Oct. 1974.

[21] J.H.Gilchrist and J.B.Thomas," Power Spectral Densities of Modulated Error-Correcting Coded Sequences,"*IEEE Trans.*, Vol.COM-23, No.11, Nov. 1975.

[22] W.R.Bennett, "Statistics of Regenerative Digital Transmission," *B.S.T.J.*, Vol.37, Nov. 1958.

[23] T.Miki, H.Kasai and H.Yamaguchi," Jitter Reduction in High Speed PCM Regenerators by External Timing," *Papers of Technical Group on Communication Systems, IECE Japan*, CS 73-101, Nov. 1973.

[24] B.Hirosaki," Study of $f_0/2$-Pilot Superposed External Timing System,"*Papers of Technical Group on Communication Systems, IECE Japan*, CS 73-102, Nov. 1973.

[25] F.Harary and E.Palmer," Enumeration of Finite Automata," *Information and Control*, Vol.10, 1967.

[26] I.Kamae,"On Encoding Automata with Power Spectral Null on a Frequency Axis,"*Graduation Thesis*, Dept. of Information Science, Kyoto University, Feb. 1977.

[27] F.Harary *et al.*,"Cospectral Graphs and Digraphs,"*Bull. London Math. Soc.*, Vol.3, 1971.

[28] S.Yajima, S.Yoshida and I.Kamae,"A Catalog of Encoding

Automata with Power Spectral Null on a Frequency Axis," *Papers of Technical Group on Automaton and Language, IECE Japan*, AL 77-12, May 1977.

[29] F.R.Gantmacher, *The Theory of Matrices*, II , New York, Chelsea Publishing Co., 1959.

[30] R.Ash, *Information Theory*, John Wiley & Sons, Inc., New York, p.209, 1965.

[31] E.R.Berlekamp, *Algebraic Coding Theory*, McGraw Hill, New York, 1968.

[32] Y.Yoshida and S.Yajima, "On the Improvement of Physical Binary Random Numbers and Its Effectiveness," *Information Processing in Japan*, Vol.11, 1971.

[33] W.R.Bennett and J.R.Davey, *Data Transmission*, McGraw Hill, New York, 1965.

[34] T.Nakagawa and S.Yajima," Implementation of LABOLINK Communication Control Programs by an Extension of Multi-User BASIC,"*Papers of Technical Group on Electronic Computer, IECE Japan*, EC 76-84, Mar. 1977.

[35] S.Yajima, Y.Kambayashi, S.Yoshida and K.Iwama," New Interfaces for Computer Communication,"*Proc. Pacific Area Computer Communication Network System Symposium*, Aug. 1975.

[36] W.H.Kautz, "Fibonacci Codes for Synchronization Control," *IEEE Trans.*, Vol.IT-11, Apr. 1965.

[37] T.Hasegawa, "A Coding Scheme for Fibonacci Code," *Trans. IECE Japan*, Vol.52-C, Feb. 1969.

[38] H.Tominaga," Recurrence Progression Codes for Band Compression," *Trans. IECE Japan*, Vol.54-A, Apr.1971.

IECE : The Institute of Electronics and Communication Engineers

IPS : Information Processing Society

LIST OF PUBLICATIONS AND TECHNICAL REPORTS

PUBLICATIONS

(1) S.Yoshida and S.Yajima, "On the Relation between an Encoding Automaton and the Power Spectrum of Its Output Sequence," *Trans. IECE Japan*, Vol.E59, No.5, May 1976.

(2) ————," Reshaping of Power Spectra of Digital Coded Signals by Restricting Inputs to Encoding Automata," *Trans. IECE Japan*, Vol.E60, No.12, Dec. 1977.

(3) S.Yajima, Y.Kambayashi, S.Yoshida and K.Iwama," New Interfaces for Computer Communication," *Proc. Pacific Area Computer Communication Network System Symposium*, Aug. 1975.

(4) ————," Optically Linked Laboratory Computer Network LABOLINK," *Proc. Tenth Hawaii International Conference on System Sciences*, Jan. 1977.

(5) ————," Labolink: An Optically Linked Laboratory Computer Network," *IEEE Computer*, Vol.10, No.11, Nov. 1977.

TECHNICAL REPORTS

(6) S.Yajima and S.Yoshida," On a Distance between Finite Automata," *Papers of Technical Group on Automaton and Language, IECE Japan*, AL 73-11, May 1973.

(7) S.Yoshida and S.Yajima, "Multilevel Transmission Codes with Spectral Nulls on the Frequency Axis and Their Encoding Automata," *Joint Convention Record of IECE Japan Kansai Branch*, G8-19, Oct. 1973.

(8) ————," Multilevel Transmission Codes with Spectral Nulls on the Frequency Axis and Their Encoding Automata," *Papers of Technical Group on Communication Systems, IECE Japan*, CS 73-108, Nov. 1973.

(9) ————," On the Neighborhood of Power Spectral Nulls of Transmission Codes," *National Convention Record of*

IECE Japan, 1394, July 1974.

(10) ———," Evaluation of Power Spectrum in the Neighborhood of Spectral Nulls,"*Joint Convention Record of IECE Japan Kansai Branch*, G7-6, Nov. 1974.

(11) ———," Consideration on Balancing of Unbalanced Transmission Code by Restricting the Set of Input Sequences," *National Convention Record of IECE Japan*, 1113, Mar. 1975.

(12) ———," Encoding Automata and Reshapings of Power Spectra of Transmission Codes by Restricting the Set of Their Input Sequences," *Papers of Technical Group on Automaton and Language, IECE Japan*, AL 75-69, Jan. 1976.

(13) ———," Power Spectra of Error Control Code Sequences," *Papers of Technical Group on Communication Systems, IECE Japan*, CS 76-124, Nov. 1976.

(14) S.Yajima, S.Yoshida and I.Kamae,"A Catalog of Encoding Automata with Power Spectral Null on a Frequency Axis," *Papers of Technical Group on Automaton and Language, IECE Japan*, AL 77-12, May 1977.

(15) S.Yajima, Y.Kambayashi, S.Yoshida, *et al.*,"Computer Connection between HITAC 8350 and PDP 11/40 and Its Applications," *National Convention Record of IPS Japan*, 201, Dec. 1974.

(16) S.Yajima, S.Yoshida and K.Tanaka,"A Sigle-Line Multi-Connector Used for a Computer Connection between HITAC 8350 and PDP 11/40," *National Convention Record of IPS Japan*, 202, Dec. 1974.

(17) S.Yajima, Y.Kambayashi, S.Yoshida, *et al.*,"Computer Connection between HITAC 8350 and PDP 11/40 and Its Applications," *Papers of Technical Group on Electronic Computer, IECE Japan*, EC 74-49, Dec. 1974.

(18) S.Yoshida and S.Yajima,"An Efficient Block Coding of Fibonacci Code," *Joint Convention Record of IECE Japan Kansai Branch*, G7-6, Nov. 1975.

(19) S.Yajima, Y.Kambayashi, S.Yoshida and K.Iwama,"Optically Linked Laboratory Computer Network LABOLINK,"*Papers of Technical Group on Computer Networks,IPS Japan*, CN7-1, Sept. 1976.

(20) ————,"Optically Linked Laboratory Computer Network LABOLINK," *National Convention Record of IPS Japan*, 326, Nov. 1976.

(21) S.Yajima and S.Yoshida,"Single-Line Multi-Connector for a Connection between Digital Devices," *System Manual*, Yajima Lab., MP75-1, Mar. 1976.